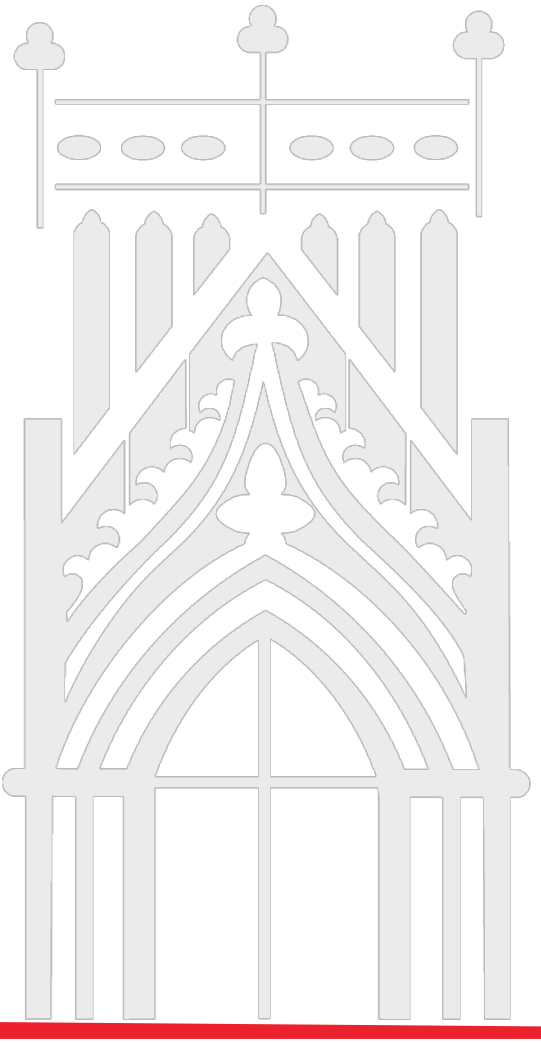
IPG Politécnico da Guarda
Polytechnic of Guarda

Mestrado em Computação Móvel

# Segurança em Bluetooth para Dispositivos Móveis



João Carlos Alfaiate

Agosto | 2014

Escola Superior
de Tecnologia e Gestão


Escola Superior de Tecnologia e Gestão

Instituto Politécnico da Guarda

# Segurança em Bluetooth para Dispositivos Móveis

Dissertação do Mestrado em Computação Móvel

Orientador: Professor Doutor José Carlos Coelho Martins da Fonseca

**João Carlos Alfaiate**

**agosto 2014**

Aos meus pais e irmã pelo incentivo recebido. Um agradecimento especial à Erica pela paciência e força que me foi dando durante este tempo.

# Agradecimentos

Quero agradecer ao meu orientador, o Professor Doutor José Carlos Coelho Martins da Fonseca, que me ensinou bastante ao longo deste percurso. As suas críticas construtivas e o rigor nas análises inspiraram-me pela área da investigação.

# Resumo

Esta dissertação fornece uma descrição detalhada sobre o Bluetooth e sobre as suas lacunas. De forma a proteger o Bluetooth, são também descritos os métodos e procedimentos para a implementação de uma futura *firewall*, deixando aqui uma porta aberta para novos investigadores.

São também estudados os tipos de ataques mais perigosos ao Bluetooth analisando o seu grau de impacto em termos de segurança. Após esta analise é proposta uma *firewall* para Bluetooth que atuará sobre o protocolo de *Radio Frequency Communication* (RFCOMM) protegendo assim o Bluetooth contra todos os ataques estudados. Uma das novidades desta *firewall* é a atribuição dos dispositivos de Bluetooth a perfis de utilização que permitem simultaneamente uma perfeita utilização destes dispositivos de Bluetooth aos quais nos ligamos, de acordo com as suas necessidades, impedindo acesso de recursos *inbound* e *outbound* desnecessarios.

Recuando vários anos atrás, as comunicações eram realizadas através de uma ligação física que utilizava cabos para a transmissão de chamadas e de dados. Com a evolução dos anos estas mesmas comunicações passaram a ser realizadas através de redes sem fios. No entanto as comunicações com redes sem fios permitem que a informação fique disponível a terceiros. De forma a fornecer segurança e manter o conteúdo das comunicações privado, foram desenvolvidos e aplicados métodos de segurança e privacidade nomeadamente nas áreas das comunicações móveis. As comunicações utilizando redes sem fios despoletou o aparecimento de várias tecnologias como o Bluetooth. Para tecnologias de curto alcance, tal como na maioria das tecnologias, a segurança na aplicação destas tecnologias, e no Bluetooth em particular, foi algo pouco explorado no seu lançamento.

Mesmo com a evolução da tecnologia de Bluetooth, a segurança foi sempre uma aposta de segundo plano até atingir a versão 2.1. Mesmo com esta versão, que tem maior foco na segurança, está provado que o Bluetooth ainda é uma tecnologia que sofre de algumas lacunas o que faz com que não seja uma tecnologia totalmente segura.

**Palavras-chave:** Bluetooth, *BlueBug*, *BlueSnarf*, *firewall*, *hacker*.

# Abstract

This dissertation provides a detailed description of Bluetooth and on its shortcomings. To protect Bluetooth, methods and procedures are also described for the implementation of a future firewall, leaving an open door for new researchers.

The most dangerous types of Bluetooth attacks are also studied and their impacts on safety were also analyzed. After this analysis, we propose a firewall that will act on the Bluetooth protocol Radio Frequency Communication (RFCOMM) thus protecting Bluetooth against all the attacks studied. One of the novelties of this firewall is the assignment of Bluetooth devices to profiles that allow a perfect use of Bluetooth according to their needs, preventing unnecessary inbound and outbound access to resources.

Stepping back several years, communication was done through a physical connection using cables to transmit calls and data. With the evolution of the years, these same communications started being conducted through wireless networks. However, communications with wireless networks allows the information to be available to third parties. In order to provide security and maintain the contents of communications private, methods for security, and privacy in particular, were developed and applied in the areas of mobile communication. Communications using wireless networks has sparked the development of other areas such as Bluetooth. Being Bluetooth a short range technology, as in most technologies, security was something unexplored in its release.

Even though Bluetooth's technology has evolved, security was always something in the background until version 2.1 came out. Even with this version, which is mainly focused on security, it is proven that Bluetooth is still a technology that still suffers from some gaps which makes it a non safe technology.

**Keywords** Bluetooth, *BlueBug*, *BlueSnarf*, *firewall*, *hacker*.

# Publicações Científicas

O desenvolvimento desta dissertação deu origem a duas publicações científicas:

- João Alfaiate and José Fonseca, "Bluetooth Security Analysis for Mobile Phones", CISTI'2012 (7th Iberian Conference on Information Systems and Technologies), Madrid, Spain, 20-23 of June 2012. O Anexo A contém este artigo científico.

  *** Citação 1**: *Marius Amund Haugen, "Kryptoanalyse og anngrep på Bluetooth". Mestrado em Tecnologia de Comunicação. Universidade Norueguesa de Ciência, Instituto de Telemática. Dezembro 2012.*

  *** Citação 2**: *Libia Malla e Diana Yacchirema, "Ataque bluebugging en dispositivos móviles Bluetooth". Escola Politécnica do Exército (ESPE), Equador. VIII Congreso de Ciencia y Tecnología ESPE 2013.*

- João Alfaiate and José Fonseca, "Bluetooth security analysis for mobile phones", INForum 2012, Universidade NOVA, Lisbon, 6-7 of September 2012. O Anexo B contém esta publicação científica.

Um outro artigo preliminar também é relevante visto tratar-se de um projeto inteiramente dedicado a redes sem fios e telecomunicações com a utilização do *Global Positioning System* (GPS) que hoje em dia é utilizado com o Bluetooth para comunicação, o que vai ao encontro desta dissertação:

- José Monteiro, João Alfaiate, Carlos Carreto, Tiago Camilo, Luís Tenedório, and Paulo Vieira, “Autonomous Robot GPS Oriented”, Engenharia 2009 (5th Conference of Innovation and Development) Covilhã, Portugal , 25-27 of November, 2009.

# Índice

# ÍNDICE DE FIGURAS

# ÍNDICE DE TABELAS

# SIGLAS

| Acrónimo | Significado em Inglês | Significado em Português |
|---|---|---|
| 1G | First Generation | Primeira Geração |
| 2G | Second Generation | Segunda Geração |
| 3G | Third Generation | Terceira Geração |
| 3GPP | 3rd Generation Partnership Project | Projeto de Parceria de Terceira Geração |
| 4G | Fourth Generation | Quarta Geração |
| ACL | Asynchronous Connectionless | Sem Ligações assíncronas |
| AFH | Adaptive Frequency Hopping | --- |
| AFRL | Air Force Research Laboratory | Laboratório de Investigação da Força Aérea |
| API | Application Programming Interface | Interface de Programação de Aplicativos |
| BCC | Bluetooth Control Center | --- |
| BT | Bluetooth | Bluetooth |
| BTSPP | Bluetooth Serial Port Profile | --- |
| CAGR | Compound Annual Growth Rate | Taxa Anual de Crescimento Combinado |
| CPU | Central Processing Unit | Unidade Central de Processamento |
| CVV2 | Credit Verification Value 2 | Código de Verificação (CVV) |
| DARPA | Defense Advanced Research Project Agency | --- |
| DPI | Deep Packet Inspection | Inspeção Profunda de Pacotes |
| DMZ | Demilitarized Zone | Zona Desmilitarizada |
| DoS | Denial of Service | --- |
| EDGE | Enhanced Data rates for Global Evolution | |
| EDR | Enhanced Data Rate | Taxa Transferência de Dados Melhorada |
| ETSI | European Telecommunications Standards Institute | Instituto Europeu de Normas das Telecomunicações |
| FTP | File Transfer Protocol | Protocolo de Transferência de Ficheiros |
| GCF | Generic Connection Framework | --- |
| GPRS | General Packet Radio Service | --- |
| GPS | Global Positioning System | Sistema de Posicionamento Global |
| GSM | Global System for Mobile Communications | Sistema Global para Comunicações Móveis |
| GUI | Graphical User Interface | Interface Gráfica do Utilizador |
| HCI | Host Controller Interface | --- |
| HID | Human Interface Devices | Interfaces Homem Máquina |
| HTTP | HyperText Transport Protocol | --- |
| I&D | Research and Development | Investigação e Desenvolvimento |
| IDS | Intrusion Detection Systems | Sistema de Deteção de Intrusões |

| | | |
|---|---|---|
| IPS | Intrusion Prevention System | Sistema de Prevenção de Intrusões |
| IPSec | Internet Protocol Security | Segurança do Protocolo de Internet |
| IrDA | Infrared Data Association | --- |
| IrOBEX | Infrared Object Exchange | --- |
| ISM | Industrial Scientific and Medical | Industriais, Científicos e Médicos |
| J2ME | Java 2 Micro Edition | --- |
| JABWT | Java APIs for Bluetooth Wireless Technology | --- |
| JSR-82 | Java Specification Request | --- |
| L2CAP | Logical Link Control and Adaptation Protocol | --- |
| LAN | Local Area Network | Rede Local |
| LMP | Link Manager Protocol | --- |
| LTE | Long Term Evolution | --- |
| MAC | Media Access Control | --- |
| NAT | Network Address Translation | Tradução de Endereços de Rede |
| NGFW | Next-Generation Firewall | Firewall de Última Geração |
| NIST | National Institute of Standards and Technology | Instituto Nacional de Normas e Tecnologia |
| OBEX | Object Exchange | --- |
| OPP | OBEX Push Profile | --- |
| OSI | Open Systems Interconnection | Interconexão de Sistemas Aberto |
| PAN | Personal Area Network | Rede Pessoal |
| PDA | Personal Digital Assistance | --- |
| PIN | Personal Identifcation Number | Número de Identificação Pessoal |
| QoS | Quality of Service | Qualidade do Serviço |
| RF | Radio Frequency | Radio Frequência |
| RFCOMM | Radio Frequency Communication | Comunicação Rádio Frequência |
| RS-232 | Recommended Standard 232 | Padrão Recomendado 232 |
| SCO | Synchronous Connection Oriented | Ligação Orientada Sincronizada |
| SDDB | Service Discovery Database | --- |
| SDP | Service Discovery Protocol | Protocolo do Serviço de Procura |
| SIG | Special Interest Group | Grupo de Interesse Especial |
| SMS | Short Message Service | Serviço de Mensagens Curtas |
| SO | Operating System | Sistema Operativo |
| SPI | Stateful Packet Inspection | --- |
| SPP | Serial Port Profile | Perfis de Porta Serie |
| SQL | Structured Query Language | Linguagem Estruturada |
| SSL | Secure Socket Layer | Camada de Ligação Segura |
| SSP | Secure Simple Pairing | --- |
| TCP | Transmission Control Protocol | Protocolo de Controlo de Transmissão |
| TCS | Telephony Control Protocol Specification | --- |
| TCS-BIN | TCS Binary | --- |
| UMTS | Universal Mobile Telecommunication System | Sistema Universal de Telecomunicações Móveis |

| | | |
|---|---|---|
| URL | Uniform Resource Locator | --- |
| UUID | Universally Unique Identifier | Identificador Único |
| vCards | Business cards | Cartão de Visita |
| VoIP | Voice over Internet Protocol | Protocolo de Voz através de Internet |
| WAF | Web Application Firewall | Firewall para Aplicações Web |
| Wi-Fi | Wireless Fidelity | Fidelidade Sem Fios |
| WLAN | Wireless Local Area Network | Redes Locais sem Fios |

# 1. INTRODUÇÃO

## 1.1. Motivação

Os telemóveis estão a tornar-se omnipresentes na vida das pessoas. Evoluíram de simples dispositivos que serviam meramente para telefonar e enviar mensagens de texto, mais conhecido como *Short Message Service* (SMS), até minicomputadores. Hoje em dia os telemóveis são capazes de navegar a Internet, ler e enviar *emails*, editar documentos, realizar cálculos complexos, executar aplicações semelhantes às dos computadores pessoais, sincronizar calendários e lista de tarefas, tirar fotografias, fazer vídeos, e muito mais. O mercado dos telemóveis cresceu de tal modo que os seus preços foram sendo reduzidos, tornando-se bastante acessíveis, o que permitiu a sua expansão em termos mundiais. Por norma, cada nova versão disponível é mais barata, mais rápida, e mais leve que as versões anteriores.

Desde o aparecimento dos telemóveis, o seu tamanho diminuiu, mas recentemente (em 2007) uma nova vaga apareceu, contendo menos botões físicos mas com um ecrã maior e *touch* para acomodar o mesmo tipo de informação que pode ser visualizado através de um computador portátil. O telemóvel revolucionador neste campo foi o iPhone com o seu primeiro lançamento no mercado em 2007 [*webdesignerdepot*, 2009].

A grande adoção dos telemóveis permitiu novos desenvolvimentos nesta área, nomeadamente em termos de comunicação. Uma das tecnologias é o Bluetooth, inicialmente criado para substituir os cabos de *Recommended Standard 232* (RS-232). O Bluetooth serve para partilhar contactos, comunicar com dispositivos de mãos livres e criar redes locais para diversos fins tais como ligar um computador a uma impressora, ouvir música de um portátil utilizando auscultadores sem fios e até formar uma rede que interligue vários dispositivos para uso doméstico (ou até mesmo empresarial) como uma rede com vários computadores.

O crescimento do mercado destes dispositivos com Bluetooth permitiu o aparecimento de novos nichos de mercados onde podem ser utilizados. Evidências destes mercados podem ser encontradas na indústria automóvel, no exército, marketing, em tarefas diárias como comércio eletrónico e banco eletrónico. Como é habitual, sempre que uma tecnologia nova como o Bluetooth é massificada, também atrai mentes maliciosas que procuram aproveitar as suas fraquezas a seu favor.

Mesmo sem terem acesso físico aos dispositivos, os *hackers* explorando o Bluetooth conseguem obter dados sensíveis lá armazenados, sendo estes facilmente transacionados na Internet. Estes ataques são críticos porque permitem obter todo o tipo de informação como o SMS, a lista telefónica, e dados do calendário. Os *hackers* conseguem efetuar estes ataques sem serem detetados e sem deixarem rastos que os possam identificar. De facto, a informação existente nos telemóveis tem grande importância ao ponto de haver diversos ataques explorando modos diferentes de obtenção a estes dados. Um exemplo é o *BlueBug* que permite envio de comandos AT (ou *Hayes commands* que permite executar um conjunto alargado de operações de comunicação) tornando possível obter informação do telefone móvel bem como efetuar uma chamada telefónica, enviar e ler mensagens de texto (SMS), aceder e modificar a lista telefónica, reencaminhar chamadas, ligar-se a outras redes sem fios, e mudar de operadora.

## 1.2. Objetivos

O objetivo deste trabalho consiste em estudar vulnerabilidades da segurança do Bluetooth, e propor o desenho de uma solução para proteger contra as vulnerabilidades na segurança do Bluetooth bem como proteger contra ataques que possam explorar outras lacunas de segurança ainda por descobrir.

Os objetivos específicos deste trabalho são os seguintes:

- Estudo do Bluetooth e a dimensão desta tecnologia na vida das pessoas. O Bluetooth tem sido uma das tecnologias que mais cresceu nesta década devido à sua fácil utilização e porque é um componente eletrónico que se encontra em muitos e diversos objetos do dia-a-dia. O crescente aumento dos dispositivos móveis, dos quais os

telemóveis são líder, também ajudou a que o Bluetooth se tornasse um dispositivo universalmente utilizável.

- Estudo dos protocolos do Bluetooth e das respetivas *Application Programming Interface*s (APIs). O Bluetooth é um dispositivo de pequena dimensão mas contém vários protocolos de comunicação para diversos fins. O objetivo é compreender a estrutura dos protocolos do Bluetooth focando nos mais importantes, tais como o *Logical Link Control and Adaptation Protocol* (L2CAP), *Radio Frequency Communication* (RFCOMM) e o *Object Exchange* (OBEX).
- Propor uma solução para proteger o Bluetooth contra as vulnerabilidades encontradas. Para combater estes ataques é proposto uma solução baseada numa *firewall* e em perfis de utilização. O Bluetooth sendo uma tecnologia de comunicação sem fios, está vulnerável a ataques por terceiros bem como está exposto a vírus e *malwares* que crescem dia após dia visto o mercado dos dispositivos móveis ser um campo ainda em franca expansão.

## 1.3. Estrutura do Documento

Este trabalho encontra-se estruturado em cinco capítulos dos quais o primeiro é composto por esta introdução ao trabalho.

No segundo capítulo é apresentado o conceito e funcionamento dos telemóveis, o tipo de comunicações mais frequentes e a utilização e importância do Bluetooth nos dias de hoje.

O terceiro capítulo explora as funcionalidades do Bluetooth bem como a sua segurança. Os ataques, a análise dos ataques, as metodologias aplicadas no estudo bem como a execução de um ataque é demostrado neste capítulo.

O quarto capítulo propõe uma *firewall* para Bluetooth para vários perfis de utilização de forma a proteger esta tecnologia contra não só nos tipos de ataques estudados mas também outros semelhantes. Este capítulo descreve em pormenor os métodos e classes que devem ser aplicados no desenvolvimento desta mesma *firewall*.

Finalmente, o quinto capítulo contém as conclusões gerais deste trabalho, analisa os seus principais resultados, e apresenta algumas perspetivas de desenvolvimentos futuros.

## 1.4. Convenções de Formatação

A Tabela 1 mostra as convenções de formatação utilizadas nesta dissertação.

**Tabela 1.** Convenções

| Convenção | Descrição |
|---|---|
| *Itálico* | Termos técnicos não traduzidos |
| **Bold** | Notas |
| `Courier New` | Código, comandos |

# 2. FUNCIONAMENTO DOS DISPOSITIVOS MÓVEIS

O componente que faz o telemóvel comunicar com outros dispositivos do mesmo tipo chama-se módulo de rádio. O módulo de rádio foi inventado em 1895 [*R. W. SIMONS*, 1996] e em 1899 a primeira mensagem oficial foi enviada de um navio da marinha norte Americana para terra [*Naval Education And Training*, 1998].

Mas o primeiro telemóvel que realmente faz uso do seu nome e que era fácil de transportar, foi criado por Martin Cooper (engenheiro na Motorola) em 3 de Abril de 1973, ou seja, há 41 anos atrás e pesava cerca de 800 gramas. Embora tenha sido criado nos anos 70, o ano de lançamento foi em 1983 com o modelo *DynaTAC 8000X* e custava 3,500 *dollars*. A Figura 1 ilustra um exemplo [*BBC News TECHNOLOGY*, 2013].

**Figura 1.** Primeiro telemóvel comercializado pela Motorola.

A forma como o módulo de rádio funciona é transformar sinais elétricos em som e vice-versa. Os sinais são enviados e recebidos utilizando a rede móvel por ondas de rádio. Estas ondas de rádio são designadas por radiação eletromagnética.

Desde a sua primeira aparição, o telemóvel deixou de ser um dispositivo que servia unicamente para comunicação de voz para ser um dispositivo com diversos tipos de comunicação. Estas comunicações dividem-se em duas: voz e dados.

Embora a maioria da comunicação por voz seja feita utilizando a rede *Global System for Mobile Communications* (GSM), é possível comunicar por voz utilizando a rede dados como o *Universal Mobile Telecommunication System* (UMTS) e o *Long Term Evolution* (LTE). Embora o GSM seja focado mais para a comunicação de

voz, é também possível comunicar com dados mas as velocidades são bastante inferiores. Em GSM o *General Packet Radio Service* (GPRS) atinge 40 Kbps em *downlink* e 14 Kbps em *uplink*. No entanto, com a evolução do GSM, surgiu também o *Enhanced Data rates for Global Evolution* (EDGE) com velocidades de 1.3 Mbps em *downlink* e 653 Kbps em *uplink* [*3gpp, Mobile Broadband Standard*]. Há que realçar que o telemóvel terá que estar equipado com antenas adequadas para cada uma destas tecnologias de forma a poder comunicar, ou seja, antenas próprias para GSM, UMTS e LTE.

A forma de comunicação de voz e dados num telemóvel não está limitada às três tecnologias referidas acima. O *Wireless Fidelity* (Wi-Fi) e o Bluetooth são outros meios de comunicação que são utilizados. O Wi-Fi tem a vantagem de poder utilizar um *hotspot* de rede sem fios para efetuar comunicação tendo acesso à Internet ou utilizando um programa de *Voice Over IP* (VoIP), como é o caso do *Skype*, para comunicação por voz. O Bluetooth por outro lado está dependente dos meios de comunicação fornecidos pelo telemóvel para comunicação, por exemplo, uma chamada de voz do telemóvel pode ser transmitida pelo Bluetooth utilizando um sistema auricular. O modo de comunicação em que o Bluetooth não necessita de outra antena de uma outra tecnologia dentro do mesmo telemóvel, será utilizando uma rede de dispositivos de Bluetooth. A grande vantagem do Bluetooth em relação ao Wi-Fi para comunicação é o facto de consumir menos energia. Todos estes dispositivos mencionados têm algo em comum, comunicam cada um com um módulo de rádio.

A verdade é que o telemóvel foi uma grande revolução em termos de crescimento. Em 2012 a *International Telecommunication Union* elaborou um relatório que mencionava que havia seis mil milhões de subscritores de telemóveis em todo o mundo [*BBC News TECHNOLOGY*, 2012]. Comparativamente aos sete mil milhões de pessoas no mundo vemos que os telemóveis têm uma grande taxa de penetração global.

Hoje em dia os telemóveis agregam vários componentes de comunicação de forma a que seja possível ter num só dispositivo funcionalidades como comunicação por voz e dados (GSM, UMTS e LTE), Wi-Fi, Bluetooth e *Global Positioning System* (GPS).

Ao ter telemóveis que num só aparelho agrega todos estes componentes, facilita a que os utilizadores possam em qualquer instante utilizar estas funcionalidades, e daí o crescimento da utilização em grandes massas como é o caso do Bluetooth.

O mercado dos telemóveis cresce com bons resultados trimestrais nas cinco maiores marcas. A Figura 2 mostra a comparação entre o terceiro trimestre do ano 2012 e 2013 do volume de telemóveis expedidos mundialmente [*IDC - Press Release*, 2013]. O gráfico mostra que todas as marcas, excepto a Nokia, aumentaram as vendas. A Nokia (recentemente comprada pela Microsoft) mesmo tendo uma diminuição nas vendas, continua destacado na terceira posição.

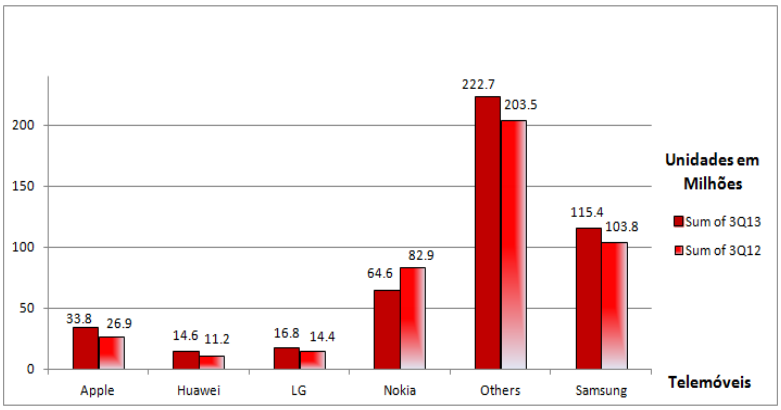



**Figura 2.** Volume de telefones móveis vendidos no 3Q 2013.

Comparando com os dados de 2010 mostrados na Figura 3 [*IDC - Press Release*, 2010], todas as marcas aumentaram as vendas. A grande diferença é que a Nokia passou da primeira posição para a terceira enquanto os *Others* destacaram-se na primeira posição duplicando o número de unidades vendidos em relação à Samsung. Outra constatação é o crescimento de marcas como o LG (multinacional sul-coreana) e a Huawei (multinacional Chinesa) que entram assim num mercado em que no passado tinham pouca expressão.

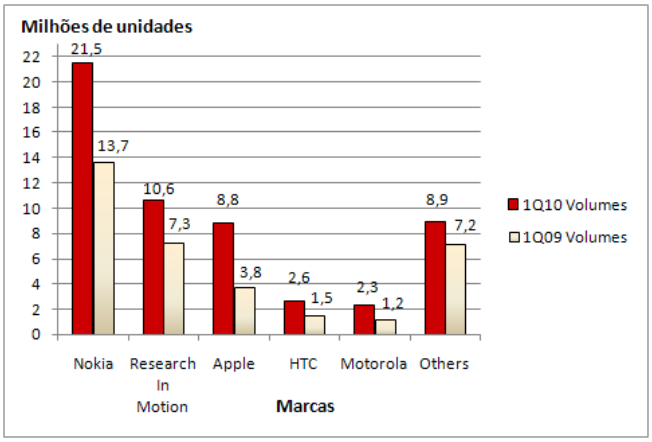



**Figura 3.** Volume de telefones móveis vendidos no 1Q 2010.

Em 2008 o valor mundialmente estimado de telemóveis era de 4.1 mil milhões [*UN News Center*, 2008]. Estima-se que em 2015 a Índia atinja mil milhões de utilizadores [*Global Thoughtz Mobile*, 2009 e *Dawn*, 2009]. A China por outro lado poderá só atingir este valor em 2020 [*Li Weitao - CHINA daily*, 2006].

## 2.1. Utilização de Bluetooth

Os telemóveis tornaram-se largamente utilizados como telefones nos anos 80, mas desde então e com a chegada de serviços digitais, houve várias mudanças. As redes de comunicações iniciais eram baseadas em sinais analógicos, também designado por tecnologia de redes móveis de Primeira Geração (1G). Esta tecnologia de 1G foi substituída pelas seguintes versões que utilizam sinais digitais e que fornecem maior velocidade, cobertura, qualidade de sinal, e serviços como Internet de alta velocidade.

As comunicações digitais começaram com a tecnologia de Segunda Geração (2G) em 1991, passando depois para a implementação da Terceira Geração (3G) para colmatar a lacuna na velocidade de transmissão de dados. O 3G foi estandardizado em 1998 e lançada com a *release 4* (Rel-4) em 2001 [*Erik Dahlman et al.*, 2008]. Não é uma surpresa ver telemóveis tornarem-se ubíquos no mundo moderno, devido às redes digitais mais eficientes e o aumento das velocidades. A Quarta Geração (4G) é intitulada como *Long Term Evolution* (LTE) e foi testado com sucesso em Boston e Seattle [*boston.com*, 2009] e implementada em 2010 [*GIGAOM*, 2010]. O 4G promete grandes melhorias como velocidade Gigabit e transmissão de *single carrier*. O LTE é

capaz de fornecer vídeo em tempo real para os dispositivos móveis, o que é um grande avanço em relação à tecnologia anterior que é o 3G. A grande exigência por redes móveis melhoradas como o LTE faz com que haja um *boom* de dispositivos móveis que irão utilizar essas mesmas redes. É estimado que o número de utilizadores de LTE atinja os 1000 milhões em 2016 [*cnet*, 2013], o que revela uma enorme exigência por este tipo de acesso à Internet e que certamente irá estar presente nos dispositivos móveis (como o telemóvel ou *tablets*).

Hoje em dia os telemóveis incorporam *hardware* como o GPS, uma antena Wi-Fi e Bluetooth. O GPS e o Wi-Fi são utilizados para localização de posicionamento e para acesso de *Local Area Network* (LAN). O Bluetooth consegue interagir facilmente com outros dispositivos de Bluetooth. A tecnologia Bluetooth começou por ser incluída nos telemóveis, mas com os anos alastrou por um leque de outros dispositivos como é mostrado na Tabela 2 [*Bluetooth SIG, History of Bluetooth Technology*].

**Tabela 2.** Objetos com Bluetooth

| Ano | Objeto |
|---|---|
| 2000 | Telemóvel, Carta PCI, Rato, Auscultadores |
| 2001 | Impressora, Portátil, Kit mãos livres para automóvel |
| 2002 | Combo de rato e teclado, Recetor de GPS, Câmara digital |
| 2003 | MP3 |
| 2004 | Auscultadores em stereo |
| 2005 | Óculos de Sol |
| 2006 | Relógio, Moldura para fotos, Relógio despertador com radio |
| 2007 | Televisão |

A tecnologia de Bluetooth foi desenvolvida para resolver o problema de comunicação por voz a curta distância de baixo consumo de energia, por exemplo, comunicação de voz com mãos livres. No entanto, em apenas sete anos o Bluetooth

tornou-se numa tecnologia madura, inserido na maior parte de objetos tecnológicos utilizados nas nossas tarefas diárias e com uma variedade de perfis de utilização.

As comunicações existentes são um *standard* em quase todos os dispositivos móveis. Os tipos de comunicação mais frequentes [*GSM Arena*] são:

- **GSM** – *Global System for Mobile Communications* tem como acrónimo GSM (originalmente, *Groupe Special Mobile*), e é a tecnologia mais utilizada no mundo para comunicação. É também conhecida por 2G e a maior fatia da comunicação é para voz. Relativamente aos dados, a comunicação é conhecido por *General Packet Radio Service* (GPRS).
- **UMTS** – O UMTS, também conhecida por 3G, é maioritariamente destinada para comunicação de dados. Ao contrário do GSM que foi conduzido pelo *European Telecommunications Standards Institute* (ETSI), a responsabilidade do UMTS foi da *3rd Generation Partnership Project* (3GPP), que é um esforço conjunto de várias organizações de *standards* para definir um sistema de comunicações móveis global de UMTS mais conhecido como 3G. O objetivo do UMTS é prover um padrão universal para as comunicações pessoais com o apelo do mercado de massa e com a qualidade de serviços equivalente à rede fixa.
- **Wi-Fi** – Utilizado por dispositivos de rede local sem fios, também conhecido por *Wireless Local Area Network* (WLAN), sendo baseado no padrão IEEE 802.11. O Wi-Fi é uma tecnologia popular que permite que um dispositivo eletrónico se ligue à Internet sem fios através de ondas de rádio. Muitos dispositivos podem usar Wi-Fi, por exemplo, computadores pessoais, jogos, smartphones, algumas câmaras digitais, *tablets* e leitores de áudio digital. Estes podem conectar-se a um recurso de rede como a Internet através de um ponto de acesso de rede sem fios. Um ponto de acesso (ou *hotspot*) tem um alcance de cerca de 20 metros em ambientes fechados e ainda mais ao ar livre. A cobertura *hotspot* pode corresponder a uma área tão pequena como um único quarto com

paredes que bloqueiam as ondas de rádio, ou tão grande como muitos quilómetros quadrados alcançados pelo uso de múltiplos pontos de acesso sobrepostos.

- **Bluetooth** – É uma tecnologia que utiliza uma rede sem fios para troca de dados em distâncias curtas (através de ondas de rádio de curto alcance) em que a partir de dispositivos fixos e móveis se constroem redes de área pessoal. Inventado pela Ericsson em 1994, foi originalmente concebido como uma alternativa sem fios para cabos de dados RS-232. O Bluetooth pode conectar-se a vários dispositivos ultrapassando os problemas de sincronização. No Bluetooth existem vários protocolos para a comunicação. Nesta pesquisa irá ser focado o *Radio Frequency Communication* (RFCOMM).

### 2.1.1. Bluetooth nos Telefones Móveis

Como o Bluetooth é um componente integrado na maioria dos telemóveis produzidos, a sua procura também está a crescer. O *Compound Annual Growth Rate* (CAGR) é a taxa de retorno de um investimento em um determinado período de tempo [*Calcular uma CAGR*], e no caso do Bluetooth cresce a 29% [*IC INSIGHTS,* 2012]. A Figura 4 mostra estes dados.

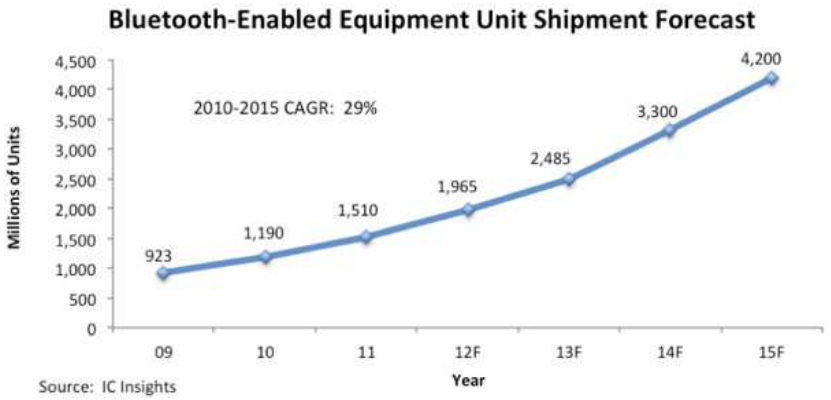



**Figura 4.** Distribuição anual do Bluetooth.

### 2.1.2. Bluetooth em e-Business

A utilização de dispositivos móveis para comércio Online está a crescer, e é esperado um aumento anual de 65% até ao ano de 2015 [*Calvin Azuri*, 2010]. Na banca isto também acontece, por exemplo, o banco da América fornece a cerca de 20 milhões de clientes a possibilidade de gerir a própria conta através de telemóveis [*Bank of America, More Than 20 Million Online Customers*]. Atualmente mais de um milhão de pequenas empresas utilizam o sistema bancário *online* [*banktech*, 2013].

### 2.1.3. Bluetooth na Indústria Automóvel

A indústria automóvel parece muito interessada na utilização de Bluetooth nos seus produtos. A seguinte lista detalha algumas funcionalidades desenvolvido especificamente para automóveis:

- O departamento de Investigação e Desenvolvimento (I&D) da Nokia apresentaram uma solução para conectar um telefone móvel com o computador a bordo da viatura, utilizando o Bluetooth [*Wireless and Mobile News*, 2009]. Esta solução disponibiliza as aplicações do telemóvel no ecrã do computador de bordo, e podem ser controladas tanto por voz ou por *touch screen*.
- A companhia Parrot SA, que fornece dispositivos sem fios para telemóveis, apresentou um computador de bordo para automóveis a funcionar com o Sistema Operativo (SO) Android, que inclui um sistema de Bluetooth para comunicação com mãos livres [*Parrot, wireless devices*].
- A Ford lançou o Ford SYNC, que é capaz de interligar com o automóvel qualquer telemóvel ou dispositivo de *media*. Este projeto foi uma cooperação exclusiva entre a Ford e a Microsoft [*Media Ford*, 2007].

### 2.1.4. Bluetooth no Exército

O exército Americano tem vários projetos que utilizam o Bluetooth como dispositivo de comunicação. Alguns exemplos desta utilização são [*Terrence Oconnor et al.*, 2008]:

- *Defense Advanced Research Project Agency* (DARPA) com uma rede sem fios para o projeto de *LANdroids.*
- *Air Force Research Laboratory* (AFRL) com o grupo de helicópteros em miniatura que estão ligados por Bluetooth.
- *Space and Naval Warfare Systems Center* com o seu robô móvel que utiliza o Bluetooth.

### 2.1.5. Bluetooth em Redes Locais

O Bluetooth é também utilizado para fazer publicidade. A cadeia televisiva CBS utiliza o Bluetooth para promover as novas séries televisivas, enviando clips de filmes às pessoas interessadas. Isto acontece no *Grand Central Station* em Nova Iorque [*Keith Regan*, 2006].

Por outro lado a Apple tem uma funcionalidade de *beacon* (dispositivo de aviso) integrado no sistema operativo iOS 7, que permite que estes dispositivos comuniquem com outros telemóveis sem que o utilizador dê autorização. Estes dispositivos já foram testados em locais públicos como no Citi Field em Nova Iorque. De acordo com o *Mashable* [*Apple Feature - Stadiums*, 2013], os *beacons* ou *ibeacon* como designa a Apple, implantados nos telefones dos utilizadores enviam dados para um servidor para que seja então exibido um mapa da área circundante ao utilizador de forma a poder ajudar as pessoas a encontrarem os seus lugares num determinado local, como por exemplo num estádio. Os *beacons* colocados dentro das lojas podem transmitir informações para os telemóveis das pessoas sobre vendas e produtos. Uma vez dentro de uma loja, os diferentes *beacons* transmitem dados sobre produtos específicos para os telefones dos utilizadores dependendo da sua localização dentro da loja [*Bluetooth beacons*, 2014].

## 2.2. Bluetooth Protocol Stack

O Bluetooth *protocol stack* está dividido em duas componentes, o Bluetooth *host* e o Bluetooth *controller* (ou módulo de rádio). Existe ainda o controlador *host* ou *Host Controller Interface* (HCI) que fornece uma interface estandardizada entre o Bluetooth *host* e o Bluetooth *controller*.

O Bluetooth *host* contém as camadas lógicas da arquitetura Bluetooth. Estas camadas incluem a implementação core do Bluetooth *stack* (pilha) e as camadas que suportam e estendem a funcionalidade do Bluetooth *stack* [*Esteban Alcorn*, 2011].

O *stack* de protocolos Bluetooth é um conjunto de camadas de protocolos que definem a funcionalidade do Bluetooth e é gerido pelo *Special Interest Group* (SIG) [*SIG - Adopted Specifications*].

A Figura 5 ilustra o diagrama de blocos do Bluetooth *protocol stack*, sendo possível visualizar os protocolos pela respetiva camada e ordem de comunicação. Dependendo do tipo de comunicação a ser utilizado, podemos ter dois ou mesmo três protocolos de comunicação, sendo estes L2CAP, RFCOMM e OBEX respetivamente.

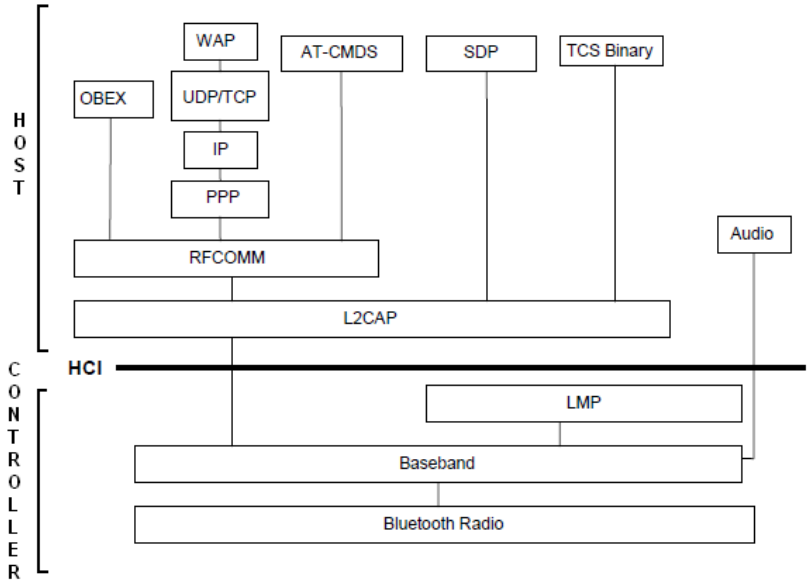



**Figura 5.** Bluetooth Protocol Stack [*JSR-82*, 2002].

O Bluetooth *stack* é constituído por diversos protocolos. Alguns protocolos como o *Link Manager Protocol* (LMP), L2CAP e o *Service Discovery Protocol* (SDP) embora sejam fundamentais para o estabelecimento da comunicação, fazem também parte dos modos de segurança do Bluetooth. Como se pode observar através da Figura 5 a maioria dos protocolos dependem do L2CAP para que haja comunicação, excepto o *Audio* que é transmitido diretamente através do *baseband* (visto que serve meramente para transmissão de voz). Existe ainda o protocolo OBEX que serve para troca de

ficheiros. O protocolo OBEX não pertence aos protocolos Core do Bluetooth, tendo sido adotado e integrado no dispositivo de Bluetooth.

Os protocolos estão agrupados por grupos e a Tabela 3 mostra como os protocolos se agrupam [*JSR-82*, 2002].

**Tabela 3.** Protocolos de Bluetooth

| Grupo de Protocolos | Protocolos no Stack |
|---|---|
| Protocolos Core do Bluetooth | Baseband, Link Manager Protocol, L2CAP e SDP |
| Protocolo substituto de cabos | RFCOMM |
| Protocolo de controlo telefónico | TCS Binary |
| Protocolos adotados | PPP, UDP/TCP/IP, OBEX, WAP |

De seguida é apresentada uma descrição sobre alguns dos protocolos mais importantes. Salienta-se no entanto, que os protocolos que irão ser focados nesta investigação são o L2CAP, RFCOMM, SDP, LMP e o OBEX visto existirem para eles *Application Programming Interfaces* (APIs) em Java para o desenvolvimento de *software* [*JSR-82*, 2002].

- **RFCOMM** – As portas série são uma das comunicações mais comuns em comutação. O protocolo RFCOMM substitui o cabo físico por uma emulação da porta série RS-232 entre dois dispositivos de Bluetooth por via de uma ligação sem fios. Esta emulação corre por cima do protocolo L2CAP, fornecendo transporte de dados a serviços nas camadas superiores. Numa ligação RFCOMM só existe uma sessão, mas cada sessão pode ter mais do que uma ligação. O número de ligações depende da implementação. Por outro lado, um dispositivo Bluetooth pode conter mais do que uma sessão de RFCOMM, desde que cada sessão esteja estabelecida a um dispositivo Bluetooth diferente.

- **OBEX** – O OBEX é um protocolo da *Infrared Data Association* (IrDA) e que foi adotado pelo Bluetooth. Este protocolo também é conhecido por *Infrared Object Exchange* (IrOBEX), mas quando aplicado ao Bluetooth, as duas letras iniciais são descartadas. Este protocolo foi definido pelo IrDA como uma alternativa ao *HyperText Transport Protocol* (HTTP) para os sistemas embebidos. Tal como o HTTP, o OBEX é um protocolo que funciona praticamente sobre qualquer camada de protocolo de transporte. Inicialmente, as implementações de OBEX eram aplicadas ao protocolo de transporte de *Infrared*, mas presentemente, as implementações funcionam também sobre ligações de *Transmission Control Protocol* (TCP), portas série e RFCOMM. Em termos do *Bluetooth Protocol Stack*, o OBEX assenta em cima do protocolo RFCOMM. Além de ter sido adotado pelo Bluetooth, as APIs do OBEX não fazem parte das APIs core do Bluetooth. No entanto estes dois grupos (packages) de APIs dependem da *package microedition* do Java. A Figura 6 ilustra esta associação.

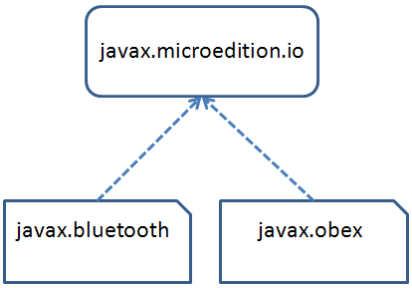



**Figura 6.** Estrutura das Packages de Bluetooth.

- **Baseband e Link Control** – Estes protocolos permitem que a ligação de Radio Frequência (RF) entre dispositivos Bluetooth seja possível. O *baseband* manipula o processamento de canais e do tempo, enquanto o *link control* manipula o acesso dos canais. Existem dois tipos de ligações físicas, síncronas ou *Synchronous Connection Oriented* (SCO), e assíncronas ou *Asynchronous*

*Connectionless* (ACL). O ACL transporta pacotes de dados enquanto o SCO suporta tráfego de áudio em tempo real.

- **L2CAP** – O L2CAP protege os protocolos da camada superior dos detalhes dos protocolos da camada inferior. Serve de multiplexador de várias ligações lógicas efetuadas pelos protocolos da camada superior, manipula a segmentação e o agregar dos pacotes de dados, e fornece *Quality of Service* (QoS) da informação.
- **SDP** – O SDP tem como função procurar serviços e características desses mesmos serviços, bem como informação sobre os dispositivos. Antes de encontrar este tipo de informação, o Bluetooth terá primeiro de encontrar os dispositivos, podendo depois procurar os serviços disponíveis nesses mesmos dispositivos.
- **LMP** – O LMP é responsável pelo estabelecimento da ligação entre dispositivos de Bluetooth, gerindo e negociando o tamanho dos pacotes do *baseband*. É também responsável pela gestão de aspetos de segurança como autenticação e encriptação ao gerar, trocar e verificar a ligação e chaves de encriptação.
- **Audio** – A especificação JSR82 não fornece APIs nem trata da informação para transmissões de áudio ou de comunicação de voz. Os dados de áudio são encaminhados diretamente de e para o *baseband*, não passando pelo L2CAP. Resumindo, o áudio não é uma camada do protocolo *stack* do Bluetooth, mas se dados de aplicações de VoIP são utilizados, os dados de áudio são transmitidos sobre o *link* de ACL.
- **TCS** – A especificação JSR82 não fornece APIs nem trata da informação para o protocolo de *Telephony Control Protocol Specification* (TCS), também conhecido por TCS *Binary* (TCS-BIN). O TCS define a sinalização do controlo de chamada para estabelecimento de chamadas de voz e de dados entre dispositivos de Bluetooth. O protocolo TCS é construído sobre o protocolo L2CAP.

## 2.3. Redes em Bluetooth

As redes em Bluetooth designam-se por *Personal Area Network* (PAN). Estas redes são pequenas redes com um número limitado de dispositivos de Bluetooth e o alcance da frequência de rádio também é limitado em média a cerca de 10 metros.

Uma PAN pode ser utilizada para diversos fins como interligar uma rede de computadores, conectar um computador ou telemóvel a uma impressora, utilizar sistemas auriculares sem fios, interligar telemóveis, e até recentemente sincronizar um telemóvel com um relógio em que se poderá fazer e receber chamadas de voz, ler SMS e notificações, e outras aplicações existentes no telemóvel utilizando um relógio criado especificamente para esse efeito [*Samsung Galaxy*].

O Bluetooth é baseado na especificação IEEE 802.11 e opera na frequência de rádio 2.4 GHz que é a frequência da *Industrial Scientific and Medical* (ISM). Esta licença é de livre utilização sendo também utilizada por outros protocolos de rádio frequência como o *Zig-Bee* e o Wi-Fi.

Uma PAN em Bluetooth pode ser criada de dois modos, uma *Piconet* ou uma *Scatternet.* As seguintes subsecções explicam cada uma destas redes.

### 2.3.1. Rede Piconet

Uma rede *piconet* consiste em dispositivos Bluetooth ligados entre si ficando um dos dispositivos como *master* e os restantes como *slave*. O *master* coordena a comunicação da *piconet* podendo enviar dados para qualquer um dos *slaves* e requisitar dados dos mesmos. O número máximo de dispositivos para uma rede *piconet* é de sete dispositivos *slave* e um dispositivo como *master*. A comunicação entre os dispositivos *slaves* terá de passar pelo *master*, e é o *master* que define o tempo de transmissão de cada *slave*.

Os *slaves* sincronizam a frequência de *hopping* utilizando o relógio do *master* e o seu endereço [*Adam Laurie et al.*, 2005].

O *hopping* é uma técnica utilizada para que a tecnologia de Bluetooth não corra o risco de causar interferência em ambientes onde outras tecnologias sem fio estão em uso como *Wireless* LAN e outras aplicações baseadas na especificação IEEE 802.11. Estas tecnologias funcionam na mesma frequência não licenciada dos 2,4 GHz, assim como Bluetooth.

A primeira geração de dispositivos de Bluetooth utilizava 79 dos 83.5 canais disponíveis na banda dos 2.4 GHz, fazendo *hopping* (saltos) por estes canais a uma média de 1600 vezes por segundo. De forma a melhorar a atuação dos ambientes do Bluetooth, uma técnica conhecida por *Adaptive Frequency Hopping* (AFH) foi introduzida pelo SIG para diminuir o impacto da interferência.

O AFH permite que o Bluetooth se adapte ao ambiente identificando pontos fixos de interferência e excluí-los da lista de canais disponíveis. Este processo de remapeamento também envolve a redução do número de canais a serem utilizados pelo Bluetooth. A especificação Bluetooth requer um conjunto mínimo de pelo menos vinte canais [*Charles Hodgdon*, 2003].

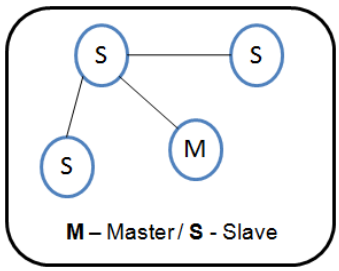



**Figura 7.** Rede Piconet.

### 2.3.2. Rede Scatternet

Uma rede *piconet* é limitada a um determinado número de dispositivos. A forma como a rede possa estender a outros dispositivos é utilizando um método que se designa por *scatternet*. Uma *scatternet* é a junção de duas ou mais redes *piconets*. Neste caso particular da *scatternet*, o *master* de uma *piconet* (letra M da Figura 8) passa a ser o *slave* de outra *piconet* (letra A da Figura 8), isto porque a *scatternet* só pode conter um master ativo de cada vez. Resumindo, um dispositivo Bluetooth só pode estar como *master* numa só *piconet*, enquanto como *slave* pode estar em várias *piconets*.

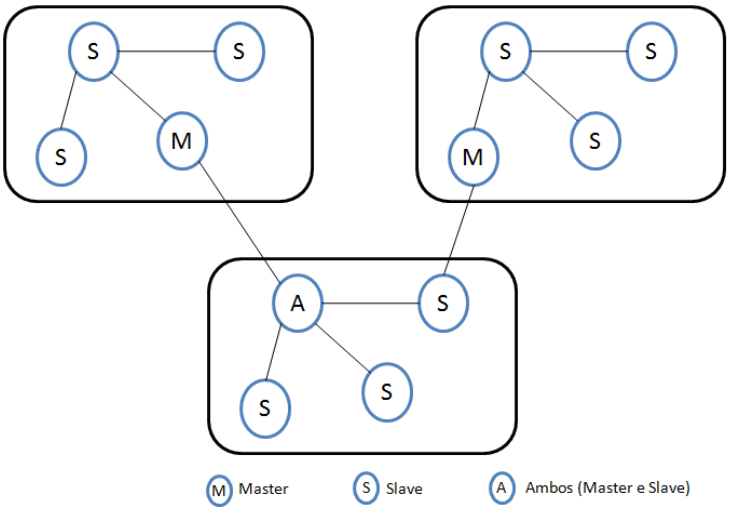



**Figura 8.** Rede Scatternet.

# 3. SEGURANÇA EM BLUETOOTH

A segurança em Bluetooth foi algo que os investigadores na altura do seu lançamento não deram muita importância. Talvez por ser uma tecnologia nova ou por ser uma comunicação de curto alcance e de uso doméstico, a atenção devida não foi prestada. Esta lacuna levou a que uma série de ataques fosse efetuado.

Esta secção explora a evolução do Bluetooth, a sua segurança e a falta dela.

## 3.1. Evolução

Desde que a primeira versão do Bluetooth chegou ao mercado em 1999, foram lançadas cinco versões. A segunda versão (versão 1.1) foi no ano de 2002, sendo a última em 2010. A Tabela 4 mostra as versões do Bluetooth bem como as principais funcionalidades e melhoramentos entre versões [*Bluetooth Specification Version 4.0*].

Através da Tabela 4, podemos verificar que até ao ano de 2004 existiu uma nova versão praticamente todos os anos. No entanto, entre 2004 e 2007 não houve novas versões. Finalmente, entre os anos 2007 e 2010 existiram três versões diferentes. Embora a versão 4 tenha sido anunciada em 2009, o seu lançamento foi em 2010.

A Tabela 4 mostra que o alvo das especificações do Bluetooth mudou ao longo dos anos. A versão 2.1 foi dedicada a aspetos de segurança e só foi lançado em 2007. As três últimas versões focaram-se no consumo de energia, que é uma enorme preocupação nos dispositivos móveis.

**Tabela 4.** Funcionalidades do Bluetooth (adaptado de [*Bluetooth Specification Version 4.0*])

| **Versões Bluetooth** | **Ano** | **Ligação mais rápida** | **SSP** | **Modo de Segurança 4** | **Correção de *Bugs*** | ***Error detection*** | **Sincronização** | **Velocidade de Transmissão de Dados** | **L2CAP** | **HCI para AMP** | **Segurança para AMP** | **Consumo de Energia** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1.1 | 2002 | | | | X | | | | | | | |
| 1.2 | 2003 | X | | | | X | X | | | | | |
| 2.0 | 2004 | | | | | | | X | | | | |
| 2.1 | 2007 | | X | X | | | | | | | | X |
| 3.0 | 2009 | | | | | | | X | X | X | X | X |
| 4.0 | 2010 | | | | | | | | | | | X |
| | | Novas Funcionalidades | | | Funcionalidades Melhoradas | | | | | | | |

A velocidade de transferência dos dados é um *bottleneck* muito comum nas telecomunicações, e tem sido modificada em quase todas as versões do Bluetooth. Neste estudo, as três principais preocupações em relação à evolução do Bluetooth são:

- Versão 2.1: Aspetos de Segurança
- Versão 3.0: *Enhanced Data Rate* (EDR)
- Versão 4.0: Baixo consumo de energia

No que refere à comunicação sem fios, existem dois fatores que prevalecem: velocidade de transferência dos dados e segurança. As seguintes subsecções descrevem estes fatores.

### 3.1.1. Velocidade de Transferência dos Dados

Desde o lançamento do Bluetooth, o mercado utiliza bastante os seus benefícios. Ao proporcionar uma taxa de transferência de dados razoável, muitos fabricantes adotaram esta tecnologia para diversos fins tais como:

- Impressoras
- Câmaras
- Telemóveis, incluindo *Personal Digital Assistance* (PDA) e *Smartphones*
- Computadores portáteis

A tabela 5 mostra a evolução da taxa de velocidade dos dados em cada versão de Bluetooth [*Bluetooth Specification Version 4.0* e *Bluetooth SIG, Basics*].

**Tabela 5.** Taxa de velocidade dos dados no Bluetooth

| Versão | Velocidade de Transmissão de Dados |
|---|---|
| 1.2 | 1 Mbps |
| 2.0 + EDR | 3 Mbps |
| 3.0 + HS | 24 Mbps |
| 4.0 | 721.2 kbps para taxas básicas (*Basic Rate*) |
| | 2.1 Mbps para EDR |
| | 24 Mbps com 802.11 AMP para operações de alta velocidade |

Observamos que o maior incremento da taxa de dados (800%) ocorreu na versão 3.0. As taxas de transferência de dados são muito importantes dependendo da utilização do dispositivo, o tipo de produto a ser utilizado e a distância a que o rádio do Bluetooth consegue alcançar.

O Bluetooth está dividido em três tipos de alcances de rádio [*Karen Scarfone et al.*, 2012]:

- Classe 1: atinge proximamente 100 metros (300 pés). Alguns exemplos são adaptadores USB e *access points.*

- Classe 2: atinge proximamente 10 metros (33 pés). Alguns exemplos são dispositivos móveis, adaptadores Bluetooth e leitores de *smart card*.
- Classe 3: atinge proximamente 1 metro (3 pés). Por exemplo, adaptadores Bluetooth.

### 3.1.2. Funcionalidades de Segurança da Versão 2.1

As funcionalidades do Bluetooth e a facilidade de utilização podem colocar em perigo a segurança do dispositivo. Por exemplo, a marinha dos Estados Unidos da América testou um método de recrutamento utilizando a capacidade de transferência de dados do Bluetooth. Durante um mês, 11 mil dispositivos móveis de Bluetooth foram detetados em 13 locais diferentes, tendo sido enviado um vídeo motivacional da marinha para 2000 desses dispositivos. Se foi possível enviar um vídeo, também seria possível enviar qualquer outro ficheiro com intenção maliciosa.

Quando a transferência de dados para recursos de terceiros é necessária, a segurança é algo que se deve ter em atenção. Com o crescimento do Bluetooth descobriu-se uma série de problemas que tentaram ser resolvidas com a versão 2.1.

A versão 2.1 do Bluetooth apresenta mais funcionalidades do que qualquer outra versão [*Bluetooth Specification Version 4.0*]. Estas funcionalidades têm impacto num grande número de aspetos relacionado com segurança:

- ***Encryption Pause and Resume***: Pausa a encriptação quando a ligação do *link key* necessita de ser alterada e quando os papéis de *master* e *slave* precisam de ser trocados. Após estas alterações a encriptação continua.
- ***Erroneous Data Reporting***: Lê e escreve o valor do parâmetro de configuração do *Erroneous Data Reporting*. Este parâmetro verifica se o controlador do Bluetooth irá fornecer dados para cada intervalo do *Synchronous Connection Oriented.*
- ***Non-Flushable Packet Boundary Flag***: Indica se o dispositivo dispõe do recurso necessário para administrar pacotes de dados HCI

ACL. Os pacotes de dados HCI ACL são usados para trocar dados entre o controlador e o *host*.

- ***Secure Simple Pairing*** (SSP): Criado para simplificar o processo de emparelhamento e melhorar a segurança do Bluetooth. Os dois principais aspetos de segurança servem para proteger contra o ataque passivo de *eavesdropping* e contra o *man-in-the-middle* ataque. O *man-in-the-middle* ataque é uma forma de *eavesdropping* (escuta de uma conversa privada) mas tem a particularidade de o atacante efetuar ligações às vítimas e transmitir mensagens entre elas, fazendo com que as vítimas acreditem que estão de facto a comunicar numa ligação privada.
- ***Sniff Subrating***: Fornece um mecanismo para reduzir o ciclo de atividades do dispositivo, incrementando a capacidade de poupança de energia do *sniff mode*. Permite também que um *host* crie uma ligação garantida ao especificar latências máximas de transmissão e receção. Isto permite que os *basebands* otimizem o desempenho de baixo consumo sem sair e voltar a entrar em *sniff mode*. O *sniff mode* aplica-se a dispositivos *salve* e é intencionado para poupar energia. Basicamente o *sniff mode* escuta (sniffa) o tráfego num modo reduzido.
- ***Security Mode 4***: É usado para o SSP.

De acordo com Andrew Lindell, criptógrafo na empresa Aladdin Knowledge Systems Ltd, o SSP parece não proteger contra o ataque *man-in-the-middle* como mencionado na especificação do Bluetooth [*Neil Roiter, Bluetooth 2.1 is easy to crack*].

Sendo o Bluetooth uma tecnologia bastante utilizada e para diversos fins, foi necessário criar vários tipos de modo de segurança de modo a proteger a ligação entre dispositivos. Com a evolução da tecnologia surgiram quatro modos de segurança que podem ser utilizados em vários cenários diferentes dependendo da versão do Bluetooth:

- ***Security Mode 1***: Um modo de segurança não muito segura. A autenticação e encriptação são ignoradas (*bypassed*). Este modo só é suportado na versão 2 e nas versões anteriores a esta.
- ***Security Mode 2***: Um modo de segurança aplicado ao nível do serviço. As medidas de segurança são iniciadas após estabelecer o LMP, mas antes de estabelecer o canal L2CAP. Este modo é suportado em todas as versões de Bluetooth.
- ***Security Mode 3***: Um modo de segurança aplicado ao nível do *link*. O dispositivo inicia medidas de segurança antes que o *link* físico esteja completamente estabelecido. Este modo de segurança só é suportado a partir da versão 2 (inclusive).
- ***Security Mode 4***: Um modo de segurança aplicado ao nível do serviço onde as medidas de segurança são iniciadas depois de estabelecer o *link*. Este modo de segurança é obrigatório a partir da versão 2.1 (inclusive) com uma exceção, em caso que o dispositivo remoto não suporte o SSP, o modo de segurança 2 é utilizado como alternativa [*Anindya Bakshi*, 2007].

O *National Institute of Standards and Technology* (NIST) considera o modo de segurança 3 como o mais robusto, devido ao requisito de autenticação e encriptação acontecer antes do estabelecimento do *link* físico [*Karen Scarfone et al.*, 2008]. Portanto, para manter *software* que utilize Bluetooth de um modo seguro, as organizações são aconselhados a utilizar o modo de segurança 3 (modo mais robusto para os dispositivos móveis), mas isto pode afetar as funcionalidades de *software* bem como a sua fácil utilização de interligação com outros dispositivos. Caso o modo de segurança 3 não seja o modo utilizado, este pode ser uma das razões para que haja diversos ataques ao Bluetooth.

## 3.2. Ataques ao Bluetooth

Esta subsecção apresenta os problemas de segurança do Bluetooth, a metodologia utilizada neste estudo, e descreve os procedimentos mais comuns utilizados pelos *hackers* para explorar as vulnerabilidades do Bluetooth.

A propagação mundial de dispositivos móveis com Bluetooth e a utilização destes em situações não previstos quando o protocolo foi desenvolvido atraiu atenções para problemas de segurança. Para combater estes problemas de segurança, a versão 2.1 do Bluetooth implementou o modo de segurança 4, que melhorou os modos de segurança de versões anteriores e suportou uma nova funcionalidade designado por SSP [*Anindya Bakshi*, 2007] que tem como objetivo facilitar o processo de emparelhamento entre dispositivos de Bluetooth com menos intervenção por parte do utilizador e garantindo segurança [*Karen Scarfone et al.*, 2008]. Embora dedicado à segurança, a versão 2.1 do Bluetooth (que corresponde ao quinto lançamento do Bluetooth, ou seja, quinta versão) ainda possui problemas de segurança por resolver. Por exemplo, é possível obter uma *password* aquando o emparelhamento dos dispositivos [*Neil Roiter, Bluetooth 2.1 is easy to crack*].

Mesmo que o Bluetooth não seja completamente seguro, ainda é utilizado em muitas situações, como descrito na secção anterior. Sendo uma ligação sem fios os utilizadores de dispositivos móveis não conseguem realmente visualizar ou sentir a ligação, e podem não estar cientes dos perigos em caso de uma violação. A maioria dos utilizadores de dispositivos móveis não conhece os danos que os ataques como o *BlueSnarf*, *BlueBug*, e o *Bluejacking* podem provocar. Um estudo realizado pela empresa de investigação InsightExpress, revelou que 73% dos utilizadores de dispositivos móveis não estão informados dos riscos de segurança que o Bluetooth pode provocar [*Lynn Tan*, 2007 e *Don Reisinger*, 2007].

Ataques a dispositivos com Bluetooth podem afetar milhões de pessoas. Isto acontece quando uma vulnerabilidade afeta um dispositivo muito difundido como o iPhone, que conta já com mais de 50 milhões de dispositivos no mundo [*João Alfaiate et al.*, 2012]. Para um equipamento tão prevalente, uma vulnerabilidade no Bluetooth foi descoberta no SDP [*Terrence Oconnor et al.*, 2008]. O ataque explora o SDP (a descoberta de serviços ativados e das suas características) para enviar uma mensagem maliciosa, permitindo ao atacante aceder ao *root shell* do dispositivo.

A descoberta de problemas de segurança do Bluetooth não é só do interesse dos *hackers*. Faz também parte das lacunas investigadas sobre esta tecnologia de modo a que possa crescer como uma opção válida de redes sem fio segura e de confiança. O

grupo Trifinite [*Trifinite Group*] é um dos maiores investidores em termos de recursos na descoberta de falhas de segurança no Bluetooth.

### 3.2.1. Metodologia no Estudo dos Ataques

A análise dos ataques ao Bluetooth é essencial para compreender quantos modos distintos existem para quebrar a segurança desta tecnologia. Por outro lado, serve também para verificar o impacto que cada ataque tem sobre o Bluetooth para que se possa correlacionar os dados e verificar a diferença entre os ataques.

Os ataques ao Bluetooth têm sido de uma certa forma pouco divulgados comparando com outros tipos de ataques informáticos que afetam computadores. Uma das razões pode ser devido ao número de utilizadores de computadores ser muito superior ao número de utilizadores de Bluetooth bem como o número de ataques aos computadores ser também muito superior. Mas para saber ao certo o número e os tipos de ataques ao Bluetooth que foram bem-sucedidos, foi necessário efetuar uma pesquisa e filtragem desses mesmos ataques. Esta secção descreve como a pesquisa foi conduzida e descreve os ataques encontrados.

As seguintes fases demonstram como o estudo sobre os ataques ao Bluetooth foi realizado:

1. A primeira fase foi compilar todo o tipo de informação disponível. Foram investigados artigos científicos, relatórios e buscas na Internet sobre segurança em Bluetooth. Alguns recursos relevantes são: o sítio de Internet do SIG [*Bluetooth SIG, History of Bluetooth Technology*], o sítio de Internet do Trifinite Group [*Trifinite Group* e *Trifinite Stuff*], o artigo *Bluetooth Network-Based Misuse Detection* de Terrence OConner e Douglas Reeves [*Terrence Oconnor et al.*, 2008], o Guide to Bluetooth Security de Karen Scarfone e John Padgette do NIST [*Karen Scarfone et al.*, 2008], entre outros.
2. A segunda fase foi a identificação dos procedimentos dos ataques. Durante a investigação, verificou-se que alguns eram mais comuns do que outros. Também foram classificados os procedimentos de ataques em dois grupos (Figura 9), Métodos e Ferramentas.

Consideramos como métodos os procedimentos que envolvem ferramentas, passos, e ações que necessitam de ser executadas. Ferramentas são *softwares* identificados que podem ter várias versões de desenvolvimento, ferramentas de configuração e administração do Bluetooth, e projetos bem-intencionados.

3. A terceira fase foi a análise dos tipos de ataques. A informação sobre as vulnerabilidades e ataques foram correlacionados para que os procedimentos de ataques pudessem ser classificados de acordo com a capacidade de afetar a vítima do ataque em termos de dados confidenciais e o controlo remoto do dispositivo.

Tal como outras vulnerabilidades de segurança (como o SQL *Injection* e XSS em aplicações Web [*José Fonseca et al.*, 2009]), existem vários modos para explorar dispositivos com Bluetooth (Figura 9). Estes modos foram divididos em ferramentas e métodos que são explicados nas seguintes secções.

A classificação dos ataques em ferramentas e métodos serve para definir como os ataques são efetuados ao Bluetooth. As ferramentas são baseadas em comandos existentes no Linux, como é o caso do *hcitool* ou o *sdptool*, e que são muito úteis na busca de informação relativamente ao Bluetooth. Por outro lado existem ferramentas de análise ao Bluetooth, sendo estas ferramentas *softwares* criados intencionalmente para profissionais na área de segurança e que ajudam a verificar lacunas que possam existir. Os métodos são procedimentos criados pelos *hackers* que conjugam as ferramentas mencionadas acima de forma a conseguirem um método eficaz de quebrar a segurança do Bluetooth. Como podemos verificar, as ferramentas são utilizadas pelos *hackers* de forma a executarem os métodos.

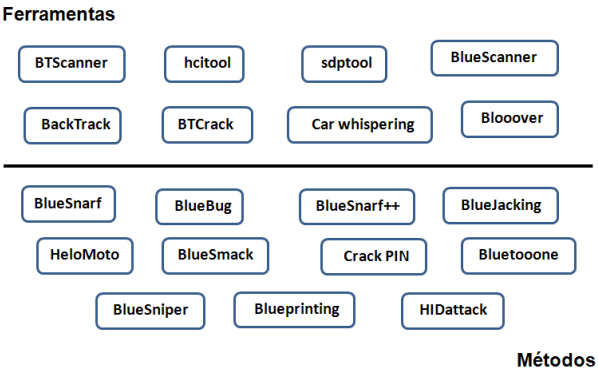



**Figura 9.** Ferramentas e métodos de ataque ao Bluetooth.

#### 3.2.1.1. Métodos

Esta subsecção descreve os métodos de ataque ao Bluetooth:

- ***BlueSnarf***: Consiste em conectar-se ao *OBEX Push Profile* (OPP) que é um protocolo de transferência de dados adotado pelo Bluetooth e que define os objetos de dados e um protocolo de comunicação que dois dispositivos podem utilizar para trocar ficheiros. Uma vez que a maioria dos casos de OPP não necessitam da autenticação do serviço, uma fraca implementação do OBEX pode ser a entrada para um ataque [*Trifinite Stuff*]. Se um *hacker* (atacante) se conectar ao OBEX e executar um pedido OBEX GET, pode obter ficheiros como a lista telefónica ou até mesmo o calendário.
- ***BlueSnarff++***: É uma melhoria do *BlueSnarf*, permitindo ao atacante acesso completo à leitura e escrita do sistema de ficheiros do dispositivo quando conectado ao OPP [*Trifinite Stuff*]. Este ataque requer que os dispositivos estejam a correr num servidor OBEX *File Transfer Protocol* (FTP) e que consigam conectar-se ao serviço *OBEX Push* sem emparelhar.
- ***BlueBug***: É o nome dado a uma vulnerabilidade presente em alguns telefones móveis, permitindo que sejam executados comandos AT remotamente em dispositivos alvo [*Trifinite Stuff*]. Um atacante que explore este método pode obter informação do telefone móvel, ou

até mesmo, ter controlo do dispositivo. Este ataque pode ser executado em segundos e permite por exemplo, efetuar uma chamada telefónica, enviar e ler mensagens de texto (SMS), aceder e modificar a lista telefónica, reencaminhar chamadas, ligar-se a outras redes sem fios, e mudar de operadora.

- ***BlueJacking***: Envia *Business cards* (vCards) de forma anónima para outros dispositivos Bluetooth através do OBEX. O *BlueJacking* consiste em enviar mensagens de texto para outros dispositivos. Estas mensagens podem conter conteúdo ofensivo o que pode intimidar e afetar psicologicamente o utilizador e até levá-lo a tomar ações indevidas.
- ***HeloMoto***: É uma combinação dos ataques *BlueSnarf* com o *BlueBug*. A origem do nome deve-se ao facto que os telefones móveis com origem desta falha de segurança eram da Motorola [*Trifinite Stuff*].
- ***BlueSmack***: É um ataque de *Denial of Service* (DoS) que faz com que os serviços do dispositivo fiquem indisponíveis [*Trifinite Stuff*]. O atacante envia um número elevado de *echo requestes* do L2CAP através do Bluetooth [*Palowireless, L2CAP*], que bloqueia o serviço do dispositivo com limitações de recursos de *hardware*. Este tipo de ataque é similar com o *Ping of Death,* que é um *ping* malicioso que bloqueia o dispositivo ou computador alvo [*Andreas Becker*, 2007].
- ***Crack PIN***: Permite que um atacante fique a conhecer o *Personal Identifcation Number* (PIN) do Bluetooth que é utilizado durante o processo de emparelhamento dos dispositivos. O processo de emparelhamento consiste em conectar dispositivos uns com os outros. Uma descrição completa em como executar este procedimento pode ser encontrada em [*Yaniv Shaked et al.*, 2005].
- ***Bluetooone***: Permite o aumento do alcance de rádio do Bluetooth [*Adam Laurie et al.*, 2004 e Adam Laurie, 2006]. Ao efetuar este procedimento, um atacante incrementa o raio de alcance do

dispositivo Bluetooth alvo, sendo capaz de perseguir mais dispositivos e sem ser detetado. Um procedimento passo a passo pode ser encontrado em [*Trifinite Stuff*].

- ***BlueSniper***: Permite executar ataques de longo alcance. Uma experiência foi realizada em Santa Monica, Califórnia, onde um dispositivo Bluetooth de classe 1 modificado alcançou 1,78 km [*Adam Laurie et al.*, 2004].
- ***Blueprinting***: É um método que obtém informação do dispositivo a ser atacado. Este método consiste em obter o identificador único do dispositivo Bluetooth que é o endereço do dispositivo Bluetooth. Ao conhecer este endereço, é possível obter o fabricante e o modelo do dispositivo, já que estes dados estão embebidos no endereço do dispositivo. Este método é útil para gerar estatísticas sobre fabricantes e modelos [*Trifinite Stuff*].
- ***HIDattack***: Utilizando o *Human Interface Devices* (HID) como o rato ou teclado do computador, este ataque conecta-se a um outro utilizador (ou dispositivo Bluetooth) passando por um dispositivo HID genuíno. A razão por que este ataque é possível deve-se à falha de implementações HID [*Terrence Oconnor et al.*, 2008].

#### 3.2.1.2. Ferramentas

Esta subsecção descreve as ferramentas de ataque ao Bluetooth:

- ***BTScanner***: É uma ferramenta baseada no *blueZ stack* do Bluetooth das plataformas Linux. O *blueZ* suporta as camadas *core* e protocolos do Bluetooth [*BlueZ*]. Com esta ferramenta, é possível detetar dispositivos que estejam ao alcance e obter informações do dispositivo como o endereço, o nome e a classe. É uma ferramenta que inicialmente foi desenvolvido para Linux, mas também existe uma versão disponível para o Windows XP chamado *BTScanner for XP* [*Pentest, Security Consultancy*].

- ***hcitool***: Foi desenvolvido para configurar ligações de Bluetooth [*hcitool,* 2002]. Esta ferramenta, tal como o *BTScanner*, é baseado no protocolo *blueZ stack* do Bluetooth. O *hcitool* é capaz de fornecer informações relevantes do dispositivo Bluetooth tais como o endereço, a classe e o nome.
- ***sdptool***: Deteta serviços através do SDP [*die.net, sdptool* e *Palowireless, SDP*]. Esta ferramenta também é baseada no protocolo *blueZ stack* do Bluetooth.
- ***BlueScanner***: Funciona no sistema operativo Windows e dispõe de um *Graphical User Interface* (GUI) ao contrário de ferramentas mencionadas em cima que se baseiam em programas de consola [*Andreas Becker,* 2007]. *BlueScanner* é um scanner de Bluetooth que deteta dispositivos de Bluetooth, obtendo o tipo de dispositivo, o endereço IP, o endereço *Media Access Control* (MAC), o código do fabricante, entre outros [*Softwar.informer*, *Bluescanner*].
- ***Blooover***: Foi desenvolvido pelo grupo Trifinite [*Trifinite Group*] e executa uma auditoria em telemóveis para verificar se são vulneráveis. O *Blooover* corre em telemóveis que funcionam com *Java 2 Micro Edition* (J2ME), como por exemplo telefones Nokia com sistema operativo Symbian ou outro telefone com o sistema operativo Android, e é capaz de executar o *BlueSnarf* e ataques limitados do *BlueBug*. Sendo uma ferramenta útil, algumas medidas de segurança foram criadas para evitar danos pessoais ou financeiros para os seus utilizadores. Por exemplo, com o *Blooover* não é possível enviar mensagens de texto (SMS), e chamadas de telefone ou reencaminhamento de chamadas só são possíveis para dispositivos que não tenham um custo associado [*Trifinite Stuff*].
- ***Backtrack***: É um *software* de segurança em Linux que é distribuído e intencionado para pessoas relacionadas com a área de segurança ou que tenham interesse neste campo [*BackTrack-linux*]. Embora este *software* seja para garantir proteção e descoberta de falhas de forma

a serem corrigidas, tal como outras tantas ferramentas de segurança, também é uma ferramenta eficaz para ataques de Bluetooth visto que consegue obter informação relativamente ao dispositivo do Bluetooth como por exemplo o endereço MAC.

- ***BTCrack***: É um *software* que força de forma bruta o PIN do Bluetooth e o *link key* (chave de ligação gerada) que são utilizados no emparelhamento de dispositivos Bluetooth. O *link key* é gerado de acordo com os modos de segurança disponíveis e tem como propósito a autenticação. O PIN do Bluetooth é um identificador numérico pessoal em cada dispositivo, chegando ao tamanho máximo de 16 bytes. Usualmente é um código com 4 dígitos [*Karen Scarfone et al.*, 2008]. Um exemplo de um procedimento de ataque pode ser encontrado em [*nruns professionals, Security Tools*].
- ***CarWhispering***: É um projeto desenvolvido para alertar fabricantes de automóveis dos perigos de implementações de Bluetooth utilizando *passkeys standard* [*Trifinite Stuff*]. A *passkey* é um parâmetro secreto utilizado para gerar e trocar o *link key*. Este ataque tem como alvo os dispositivos de áudio mãos livres, e é capaz de injetar áudio no dispositivo alvo bem como gravar áudio ao vivo desse mesmo dispositivo [*Terrence Oconnor et al.*, 2008].
- ***hcitool***: É uma ferramenta de administração do Linux e é capaz de obter informações do dispositivo como o endereço, a classe e o nome. Esta mesma informação é também obtida através do *BTScanner*.

### 3.2.2. Análise de Ataques ao Bluetooth

Esta secção descreve a análise dos procedimentos de ataque e o seu impacto nas vítimas.

#### 3.2.2.1. Métodos e Ferramentas

Durante a investigação foram encontrados oito ferramentas e onze métodos de ataque ao Bluetooth. A Tabela 6 mostra a evolução dos ataques do ponto de vista cronológico de acordo com o ano de aparecimento. A Figura 10 mostra o número de ataques por ano do ponto de vista da sua evolução. Podemos verificar que a maioria dos ataques apareceu entre os anos 2004 e 2007. Este período corresponde ao *upgrade* da versão 2.0 para 2.1 do Bluetooth, que demorou três anos a ser lançado. Depois de a versão 2.1 ser lançada no final do ano 2007, não foram encontrados novos ataques na literatura que foi pesquisada.

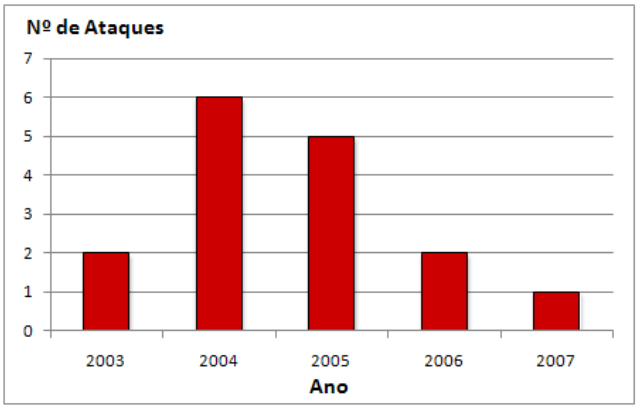



**Figura 10.** Evolução dos ataques (referencias nas secções 3.2.1.1-Métodos e 3.2.1.2-Ferramentas).

Durante a pesquisa não foram encontradas referências que provam que as vulnerabilidades existentes tenham sido totalmente resolvidas com a versão 2.1 do Bluetooth. Na verdade, existem investigadores que questionam se a versão 2.1 consegue parar todos os ataques [*Andrew Y. Lindell*, 2008]. Uma vulnerabilidade identificado como o *malware W32.Flamer* ocorreu no Bluetooth que aparentemente é o primeiro *malware* que utiliza o Bluetooth para espiar as vítimas. O NIST não aborda especificamente o *Flamer*, mas recomenda que um *software* de antivírus seja usado para bloquear o *malware* a partir de outros dispositivos Bluetooth [*William Jackson*, 2012]. Talvez esta seja umas das razões para que as diretrizes publicadas em 2008 tenham sido atualizadas em 2012, sendo estas a mitigação da vulnerabilidade para *Secure Simple Pairing* (SSP) bem como uma introdução e discussão de mecanismos de segurança e recomendações para Bluetooth v3.0 e v4.0 [*Karen Scarfone et al.*, 2012].

Mesmo que a versão 2.1 tenha implementado o modo de segurança 4 com SSP para facilitar o processo de emparelhamento entre os dispositivos Bluetooth, um atacante pode obter a *password* quando estes mesmos diapositivos estão a ser emparelhados.

Andrew Lindell mencionou que na versão 2.1 do Bluetooth, uma *password* fixa pode ser obtida com um ataque *man-in-the-middle*, independentemente do comprimento da *password*, enquanto na versão 2.0, uma *password* comprida podia prevenir o ataque [*Neil Roiter, Bluetooth 2.1 is easy to crack*]. Dá a entender que enquanto são feitos progressos em algumas áreas, noutras existe regressão.

**Tabela 6.** Ferramentas e métodos de ataque ao Bluetooth

| Procedimento | | OS | | Ataques | |
|---|---|---|---|---|---|
| | Ano | Linux | Windows | Ferramenta | Método |
| BTScanner | n.a | X | X | X | |
| hcitool | n.a | X | | X | |
| sdptool | n.a | X | | X | |
| BlueSnarf | 2003 | | | | X |
| BlueJacking | 2003 | | | | X |
| Blooover | 2004 | X | X | X | |
| BlueBug | 2004 | | | | X |
| BlueSmack | 2004 | | | | X |
| Bluetooone | 2004 | | | | X |
| BlueSniper | 2004 | | | | X |
| Blueprinting | 2004 | | | | X |
| BlueSnarf++ | 2005 | | | | X |
| HeloMoto | 2005 | | | | X |
| Crack PIN | 2005 | | | | X |
| Car whispering | 2005 | X | | X | |
| HIDattack | 2005 | | | | X |
| BackTrack | 2006 | X | | X | |
| BlueScanner | 2006 | | X | X | |
| BTCrack | 2007 | | X | X | |

Como se pode verificar na Tabela 6, existem vários modos de fazer um ataque a um dispositivo Bluetooth. Alguns procedimentos são simples e não acedem a

informação pessoal, enquanto outros procedimentos são de facto capazes de obter dados pessoais ou até mesmo obter controlo do dispositivo. A análise mostra que os três procedimentos mais recentes são ferramentas, e as duas mais recentes são especificamente para o sistema operativo Windows. Ao desenvolver este tipo de *software* para utilizadores Windows, incrementa potenciais utilizadores não relacionados aos profissionais de segurança. A ferramenta *Blooover* está classificada como funcional tanto em ambientes Linux como Windows, visto que é uma ferramenta desenvolvida em telefones móveis em que os sistemas operativos funcionam com J2ME.

#### 3.2.2.2. Impacto do Ataque

De modo a analisar melhor o impacto que os ataques em Bluetooth têm sobre as vítimas, precisamos de saber o que o atacante pode obter do dispositivo. A Tabela 7 mostra os procedimentos que conseguem obter informações dos dispositivos, sendo esta informação utilizada em vários tipos de ataque como o *BlueBug*, *BlueSnarf* e o *HeloMoto*.

**Tabela 7.** Informação sobre dispositivos obtidos através de ataques [*C Bala Kumar et al.*, Motorola 2003]

| Ferramenta | Informação do Dispositivo | | | | | |
|---|---|---|---|---|---|---|
| | **Address** Endereço MAC | **Class** Ex: telemóvel, dispositivos áudio, impressoras, câmaras, entre outros | **Name** Nome do dispositivo | **Type** Tipo do dispositivo (Ex: USB, outro) | **PIN** Código para emparelhar os dispositivos | **Services** Serviços disponíveis |
| BTScanner | X | X | X | | | |
| hcitool | X | X | X | | | |
| Blueprinting | X | | | | | |
| BlueScanner | X | | | X | | X |
| BTCrack | | | | | X | |

A Tabela 8 mostra os impactos causados pelos diferentes tipos de procedimentos de ataque.

**Tabela 8.** Impacto dos Ataques

| | Impacto | | | | | |
|---|---|---|---|---|---|---|
| **Procedimento** | **SDP** | **OBEX** | **Auditoria de segurança** | **Envio de vCard** | **Envio de comandos AT** | **Ataque DoS** |
| BTScanner | X | X | | | | |
| hcitool | X | X | | | | |
| sdptool | X | | | | | |
| BlueSnarf | | X | | | | |
| BlueJacking | | X | | X | | |
| Blooover | | | X | | | |
| BlueBug | | | | | X | |
| BlueSmack | | | | | | X |
| BlueSnarf++ | | X | | | | |
| HeloMoto | | | | X | X | |
| BTScanner | X | X | | | | |

Embora alguns procedimentos têm apenas um impacto, estes impactos podem ser perigosos porque podem remotamente aceder a dados no telemóvel bem como bloquear o mesmo, sendo estes o "Envio de comandos AT" e o "Ataque DoS" respetivamente. A seguinte lista descreve os impactos mostrados na Tabela 8:

- **SDP**: Permite a descoberta de serviços disponíveis e das suas características.
- **OBEX**: Facilita a troca de objetos binários entre dispositivos.
- ***Security Audits***: Mede avaliações técnicas de um sistema ou aplicação.
- **Envio de vCards**: Envia mensagens para outros dispositivos Bluetooth.
- **Envio de Comandos AT**: Permite controlo remoto de dispositivos de comunicação.
- **Ataque DoS**: Intencionado para tornar o dispositivo indisponível.
- **Verificação de Vulnerabilidades conhecidas**: Executa uma auditoria em telemóveis para verificar se são vulneráveis.

#### 3.2.2.3. Análise Detalhada

Os impactos dos ataques mostrados na Tabela 8 são bastante diferentes uns dos outros, sendo importante identificar qual o ataque mais crítico em termos de acesso a informação privada, ou até mesmo, a possibilidade de ter controlo sobre o dispositivo. Por exemplo, a habilidade do *BlueJacking* em enviar mensagens de texto para outros dispositivos é menos prejudicial do que a habilidade do *BlueBug* em enviar comandos AT, isto porque, embora o *BlueJacking* possa interferir e intimidar de alguma forma um utilizador com uma determinada mensagem, não consegue aceder remotamente ao telemóvel como o *BlueBug* é capaz.

Os dois procedimentos que conseguem enviar comandos AT e poder controlar remotamente o dispositivo são o *BlueBug* e o *HeloMoto*. O *HeloMoto* pode não ser um ataque com grande amplitude visto que só afeta alguns telemóveis da Motorola. O ataque do *BlueBug* parece mais perigoso, já que pode ser executado em diversos dispositivos de várias marcas. O nome das marcas não estão publicamente

disponíveis porque o grupo Trifinite, que as identificou, só revela esta informação aos fabricantes dos dispositivos [*Trifinite Stuff*].

Os ataques também podem obter dados pessoais armazenados no dispositivo móvel como fotografias, SMS, e outro tipo de dados. Isto é um caso sério, porque dados privados podem ser negociados mundialmente no *underground market* [*Symantec*, 2008]. Embora alguns dispositivos móveis permitem o armazenamento de ficheiros, as funcionalidades mais comuns presente nos telemóveis estão na Tabela 9. A Tabela 9 também correlaciona as funcionalidades com o respetivo tipo de ataque. Com o *BlueBug*, acrescendo à possibilidade do seu acesso à lista telefónica e SMS com a possibilidade de efetuar telefonemas ler e enviar mensagens, é também capaz de conectar-se com outras redes sem fios [*Trifinite Stuff*].

A Tabela 9 mostra o impacto de três ataques às funcionalidades mais utilizadas nos telemóveis:

**Tabela 9.** Impacto dos ataques na informação pessoal

| | **Dados Privados** | | | | |
|---|---|---|---|---|---|
| **Procedimento** | **Fotografias** | **Calendário** | **Lista Telefónica** | **SMS** | **leitura/escrita no sistema de ficheiros se conectado ao OPP** |
| BlueSnarf | X | X | X | | X |
| BlueSnarf++ | X | X | X | | X |
| BlueBug | | | X | X | |

## 3.1. Bluetooth Malwares

Um investigador da McAfee, Jimmy Shah, reportou um *trojan* chamado *Obad.a* que afeta o sistema operativo Android. Este *trojan* tem várias características que faz com que evite ser detetado, o que o torna difícil de detetar com o uso de práticas tradicionais de segurança [*Alastair Stevenson*, 2013]. Roman Unuchek do laboratório de Kaspersky disse em maio de 2013 que o *Obad.a* é o *trojan* móvel mais sofisticado até ao momento [*Roman Unuchek*, 2013].

O *Obad.a* consegue ainda enviar SMS para números pré-definidos e que normalmente são de serviços que têm uma taxa de utilização de valor elevado. Depois do *Obad.a* receber um comando de um servidor gerido por um *ciber* criminoso, o *malware* faz um scan a dispositivos circundantes com ligações abertas e tenta enviar uma aplicação (uma *app* perigosa) para esses dispositivos [*John P. Mello Jr*, 2013].

O *Cabir* é um *worm* (um tipo de vírus) que utiliza o Bluetooth para se disseminar e tem impacto nos telemóveis com o sistema operativo Symbian. É possível remover este *worm* utilizando o Mobile Antivírus da F-Secure [*F-Secure*].

O *Flame* é outro *malware* que consegue utilizar o Bluetooth para filtrar dados, gravar telefonemas, ou até mesmo aprender sobre a rede social do utilizador que está a ser atacado [*Mathew J. Schwartz*, 2012].

Após verificarmos que existem *malwares* que têm impacto nos dispositivos móveis, convém saber qual a marca com maior incidência. De acordo com investigadores da Cisco, quando um *malware* é destinado a comprometer um dispositivo móvel, 99% dos alvos são dispositivos Android. Os *trojans* destinados a dispositivos com plataforma J2ME ficaram em segundo lugar no ano de 2013 com apenas 0,84% de todos os *malwares* encontrados. Os *malwares* que têm como alvo dispositivos específicos somam 1,2% de todos *malwares* encontrados na Internet [*iClarified*, 2014].

Uma análise da Cisco revela também que 71% dos utilizadores de Android têm as maiores taxas de impacto com todas as formas de *malware* presentes na Internet, seguidos por utilizadores do iPhone com 14% [*iClarified*, 2014].

A Figura 11 mostra os *malwares* encontrados na Internet e a percentagem de impacto que têm sobre os dispositivos móveis.

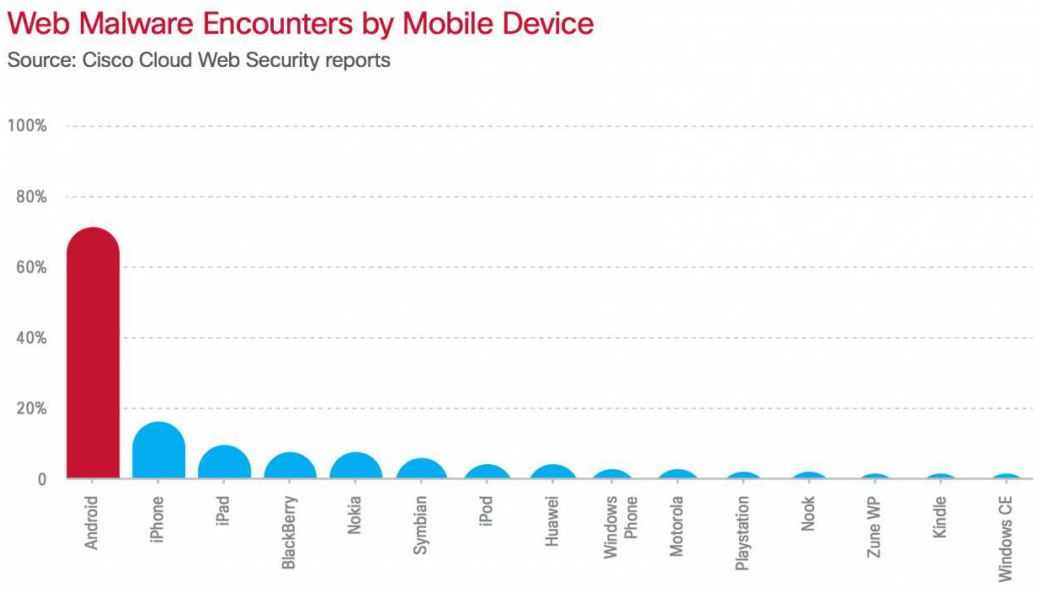



**Figura 11.** Marcas com mais impactos de malewares [*iClarified*, 2014].

Um problema grave que se nota com os estudos e análises referidos acima, é que muitos telemóveis são infetados com novos vírus e *malwares*, o que mostra que os mecanismos de defesa existentes nos telemóveis contra estes tipos de ataques são ineficazes, ou seja, será necessário que os telemóveis sejam infetados para que o problema seja depois endereçado e resolvido. Mais uma vez existe uma falha de segurança com o Bluetooth que não consegue lidar com problemas de disseminação de vírus, e que hoje em dia é o grande problema para os fabricantes de telemóveis visto ser uma área nova ao contrário dos computadores que já contam com vários anos de investigação e proteção contra este tipo de problemas.

## 3.2. Prevenção

O Bluetooth é utilizado em muitos objetos pessoais utilizados diariamente, mas o cenário mais provável de utilização é para comunicação móvel. Em alguns países é proibida a utilização de telemóveis enquanto se conduz um automóvel [*Catherine Roseberry, About.com*], mas utilizando um sistema de comunicação sem fios para comunicar com o telemóvel já é legal visto que o utilizador tem ambas as mãos livres. Muitos utilizadores utilizam o Bluetooth de uma forma regular e esquecem-se de o

desligar quando não o utilizam. Outros utilizadores esquecem-se de alterar as definições de segurança que vêm por omissão para uma mais segura.

O roubo de telemóveis permite obter informação valiosa de forma a negociar no mercado negro ou para extrair dados valiosos como informações de cartões de crédito, números de telefone e *passwords*. Esta tendência é confirmada pelo *CSI Computer Crime and Security Report* de 2009, que menciona que o roubo de portáteis e de dispositivos móveis é o segundo incidente mais relevante e que foi reportado por 42% dos inquiridos [*CSI*, 2009]. De acordo com o *Symantec Corporation* [*Symantec*, 2008] dados sensíveis têm uma procura imensa em mercados de *underground economy*. Verificaram que os dados sensíveis mais transacionados no *underground economy* são os *Credit Verification Value 2* (CVV2 – são os três ou quatro últimos dígitos que usualmente se encontram na parte de trás dos cartões de crédito) que correspondem a 23% das transações. Em segundo lugar aparecem os números dos cartões de crédito com 18% das transações. Os números de telefone aparecem em quinto lugar com 11% enquanto os PIN aparecem na sétima posição com 5% das transações.

Foi realizado um teste em Londres para analisar quantos utilizadores de Bluetooth podiam estar sujeitos a um ataque. Dos 943 dispositivos móveis identificados, 40% tinham as definições de Bluetooth por omissão, e 138 dispositivos de Bluetooth estavam vulneráveis ao ataque do *BlueSnarf* [*Kevin Streff et al.*, 2009]. Outro teste foi realizado pelo CeBIT na feira de tecnologia em Hannover. Neste teste foram detetados 1,300 dispositivos de Bluetooth, sendo que 50 destes dispositivos eram vulneráveis ao ataque do *BlueBug* [*Trifinite Stuff*]. Estes dois exemplos demonstram que o Bluetooth é frequentemente utilizado e que os utilizadores usam o Bluetooth de uma forma vulnerável. Mesmo com uma melhoria na especificação do Bluetooth, os utilizadores deveriam de seguir as melhores práticas de segurança porque estas práticas podem prevenir a maioria dos ataques [*Karen Scarfone et al.*, 2008]:

- Desconectar o Bluetooth quando não for utilizado.
- Alterar o PIN que existe por omissão.
- Emparelhar dispositivos o menos possível.
- Utilizar o Bluetooth de classe 2 ou 3 para comunicações de curta distância.

- Dispositivos de Bluetooth como auscultadores devem permanecer em modo não visível.
- Remover serviços não utilizados do dispositivo Bluetooth.

Embora os utilizadores de Bluetooth devam seguir as melhores práticas como desligar o Bluetooth quando este não está em uso, restringir configurações do Bluetooth e remover *trusted devices* quando não forem necessários, os dispositivos de Bluetooth deviam de fornecer uma barreira de segurança por norma, que proteja os utilizadores.

## 3.3. Teste de Ataque BlueBug

Os ataques descobertos e expostos por outros investigadores são importantes para o desenvolvimento deste estudo, mas foi necessário recriar um ataque para verificar a potencialidade do mesmo.

A finalidade deste teste de ataque era para verificar a possibilidade de violar as definições de segurança do Bluetooth e ver até que ponto era possível ter completo acesso remoto a um telemóvel utilizando o Bluetooth.

Sendo o *BlueBug* um dos ataques mais perigosos, este foi escolhido como teste por forma a desenvolver uma aplicação que proteja contra este tipo de ataques. O ataque do *BlueBug* é aplicado ao nível do protocolo de RFCOMM, onde controlo total do dispositivo é possível com execução de comandos AT. Assim sendo, protegendo o RFCOMM deste ataque, também protege contra outros ataques que dependem deste protocolo de acordo com as camadas definidas no *Bluetooth Protocol Stack*, ou seja, ataques utilizando o OBEX poderão também ficar protegidos, sendo o *BlueSnarf* um destes ataques.

### 3.3.1. Ligação e Acesso

O RFCOMM é uma emulação da porta série RS-232, que estabelece uma ligação sem fios entre dois dispositivos. A comunicação feita é comparada a uma ligação por *sockets* visto que os dados são enviados em forma de *stream*. De realçar que

só pode existir um *link* físico (ou sessão) entre dois dispositivos, mas dentro deste mesmo *link* pode existir múltiplas ligações [*JSR-82*, 2002].

No Bluetooth existem quatro níveis de segurança quando se efetua uma ligação a outro dispositivo Bluetooth. Estes níveis de segurança são:

- *Pairing* (emparelhamento).
- *Authentication* (autenticação).
- *Encryption* (encriptação).
- *Authorization* (autorização).

O primeiro passo quando se estabelece uma ligação entre dois dispositivos de Bluetooth é o emparelhamento. Este emparelhamento é necessário para que se possa estabelecer uma chave secreta que será depois utilizada na autenticação e encriptação. No emparelhamento, ambos os dispositivos necessitam de inserir um código PIN idêntico, que será depois utilizado para autenticação. O *Bluetooth Control Center* (BCC) é o responsável pelo emparelhamento, tendo como função o pedido e registo do PIN.

Ao contrário de muitos programas em que a autenticação é feita ao nível do utilizador, por exemplo, acesso a uma conta de *email*, ao Facebook, entre outros, a autenticação do Bluetooth não identifica utilizadores, só autentica dispositivos. Após autenticação a encriptação pode ou não ser ativada, isto porque, para que a encriptação seja ativada, ambos os dispositivos têm de ter os parâmetros de encriptação ativos (na implementação do *software*) para poderem comunicar com pacote de dados encriptados. Caso um dos dispositivos não tiver os parâmetros de encriptação ativos, a comunicação será feita sem encriptação.

A autorização é um procedimento que define se um pedido de ligação por outro dispositivo deva ser aceite. De modo a facilitar a ligação dos dispositivos de Bluetooth, e para que não se despenda de tempo e trabalho ao definir o PIN e a aceitação do mesmo, a especificação do Bluetooth criou o chamado *trusted device* que é um registo que fica no dispositivo Bluetooth sempre que um utilizador emparelhar um dispositivo novo. Este mecanismo de *trusted device* garante autorização automática quando este é pedido, sendo possível acesso a qualquer serviço existente no dispositivo. Este mecanismo é muito útil para utilizadores que usam o Bluetooth sistematicamente

para o mesmo fim, por exemplo, *kit* mãos livres para comunicação de voz dentro de um automóvel. O BCC não interfere com a autorização dos *trusted device*, mas é responsável pela lista destes mesmos dispositivos. O BCC só interfere quando um dispositivo não fizer parte desta lista.

Estes níveis de segurança estão interligados entre si. A autenticação requer emparelhamento, e a encriptação e autorização requerem autenticação.

### 3.3.2. Laboratório de Teste do BlueBug

De forma a realizar os ataques, foi necessário criar um laboratório de testes. O ataque escolhido foi o *BlueBug* porque é um ataque que permite obter controlo sobre o telemóvel, ou seja, um ataque bastante perigoso. Como o ataque do *BlueBug* depende do envio de comandos AT para o telemóvel, foi necessário recriar como o ataque podia ser executado. Era sabido que o ataque utilizava ferramentas de administração do Linux para este efeito, e também era necessário uns telemóveis como alvo. Como máquina de ataque foi utilizado um computador *desktop* com as seguintes características:

- Processador: Intel Pentium 4 a 2.4GHz
- Memória RAM: 768 MB
- SO: Linux Ubunto versão 8.04.1
- Dispositivo de Bluetooth: *Dongle* USB com a versão 2.1.

Em relação aos telemóveis alvos, ou telemóveis a serem atacados, foram utilizados dois Nokias normais e um *smartphone*. Todos os telemóveis eram de anos diferentes e também tinham versões de Bluetooth diferentes. A Tabela 10 mostra as caraterísticas de cada telemóvel.

**Tabela 10.** Telemóveis usados no teste de ataques

| Telemóvel | Ano | Versão BT | Ligação *bypassed* se BT emparelhado |
|---|---|---|---|
| Nokia 3110 classic | 2007 | v2.0 | Não |
| Nokia 6303i classic | 2010 | v2.1 | Sim |
| Samsung OMNIA II | 2009 | V2.0 + EDR | Não |

A Tabela 10, além de mostrar as características relevantes dos telemóveis usados para o teste, salienta ainda a necessidade (ou não) de aceitar manualmente (ou *bypass* automático) uma ligação de um dispositivo Bluetooth previamente conhecido.

Podemos desde já verificar uma diferença importante na forma como os telemóveis lidam com o emparelhamento de dispositivos previamente associados. Enquanto o *Nokia 3110 classic* e o *Samsung OMNIA II* precisam que o utilizador aceite (ou negue) a ligação Bluetooth, o *Nokia 6303i classic* ignora a autorização e aceita todos os *trusted devices* por omissão. É curioso verificar que este último telemóvel utiliza a versão supostamente mais segura do Bluetooth, sendo esta a versão 2.1. Se um atacante é capaz de fazer um dispositivo não autorizado ter ligação com o telefone de destino, ele será capaz de atacar o telefone a qualquer momento sem ser notado.

### 3.3.3. Descrição do Ataque BlueBug

O objetivo do ataque é estabelecer uma ligação Bluetooth usando o RFCOMM como ligação ao telemóvel de forma a enviar comandos AT (Anexo C contém uma lista de comandos AT). Para que este procedimento seja possível, o dipositivo terá de estar emparelhado com a máquina de ataque. Como hoje em dia o Bluetooth é usado para muitos fins, é usual que cada telemóvel tenha vários dispositivos de Bluetooth emparelhados o que facilita o ataque.

Supondo então que um telemóvel a ser atacado tenha pelo menos um dispositivo de Bluetooth emparelhado, o seguinte comando mesmo sendo inofensivo, permite obter toda a informação do dispositivo alvo (e funciona em qualquer telemóvel). Esta informação poderá ser utilizada em ataques:

```
Desktop# sdptool browse [MAC]
```

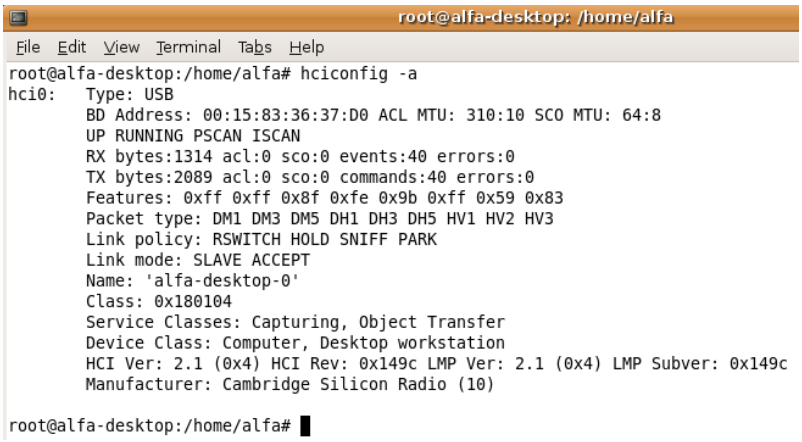
O seguinte procedimento exemplifica passo a passo como ligar o servidor a outro dispositivo de Bluetooth e configurar o ataque:

1. Abrir uma janela de terminal.
2. Inserir o comando que procura dispositivos Bluetooth de forma a obter o endereço MAC:

`Desktop# hcitool scan`

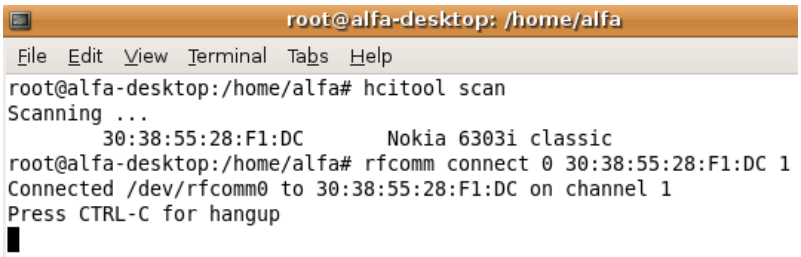

3. Inserir o comando que mostra as características do Bluetooth da máquina de ataque. Neste contexto serve para verificar que o Bluetooth da máquina de ataque está a funcionar:

   `Desktop# hciconfig –a`



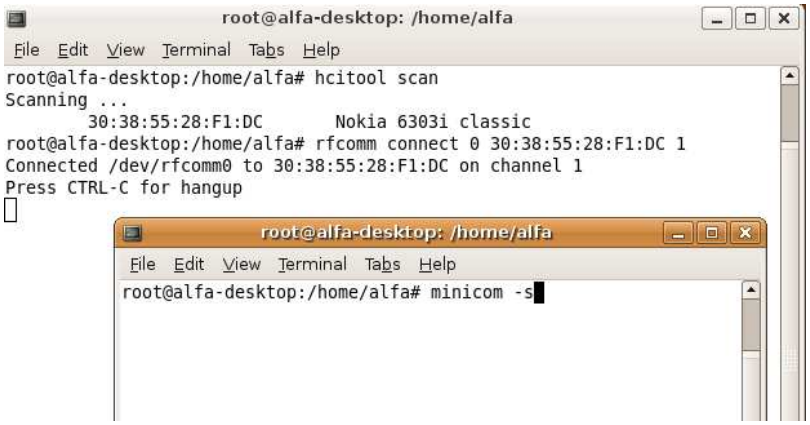

4. Inserir o comando que efetua a ligação através de RFCOMM da máquina de ataque ao dispositivo alvo:

   `Desktop# rfcomm connect 0 [`*`Mac dispositivo alvo`*`] 1`

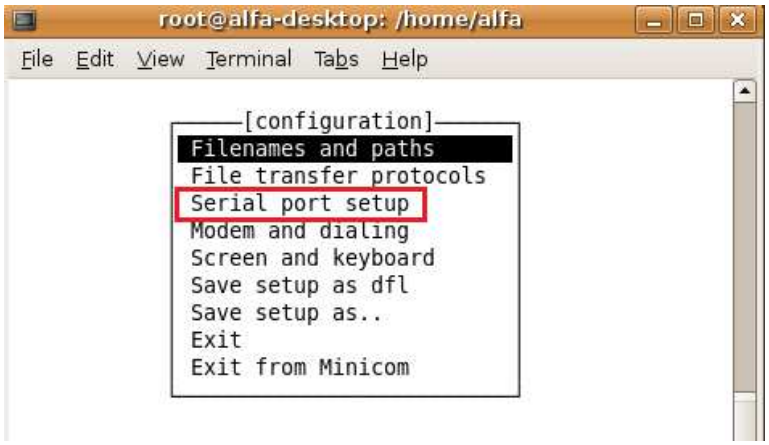



5. Abrir outra janela de terminal e Inserir o comando que serve para entrar no modo de configuração do *minicom*.

```
Desktop# minicom –s
```

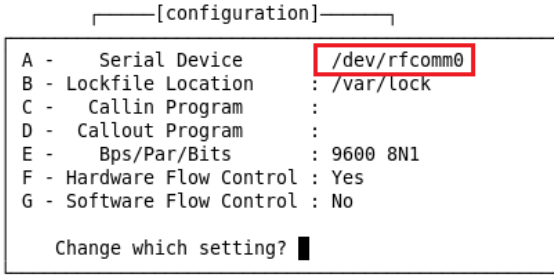



6. Selecionar **Serial port setup** e pressionar **Enter**. Irá abrir outra janela de configuração de parâmetros.




7. Configurar a opção **A – Serial Device** de forma a ter `/dev/rfcomm0`.

   Exemplo: `A- Serial Device : /dev/rfcomm0`.




8. Retroceder à janela anterior e escolher a opção **Save setup as dfl**.

Após efetuar os passos mencionado em cima o telemóvel fica preparado para receber comandos AT, o que indica que o ataque está pronto a ser executado.

Com o *Samsung OMNIA II* o RFCOMM faz a ligação, mas quando se liga o *minicom*, a ligação RFCOMM vai abaixo. A Figura 12 e Figura 13 ilustram esta situação.


**Figura 12.** Ligação ao Samsung Omnia II através da RFCOMM.

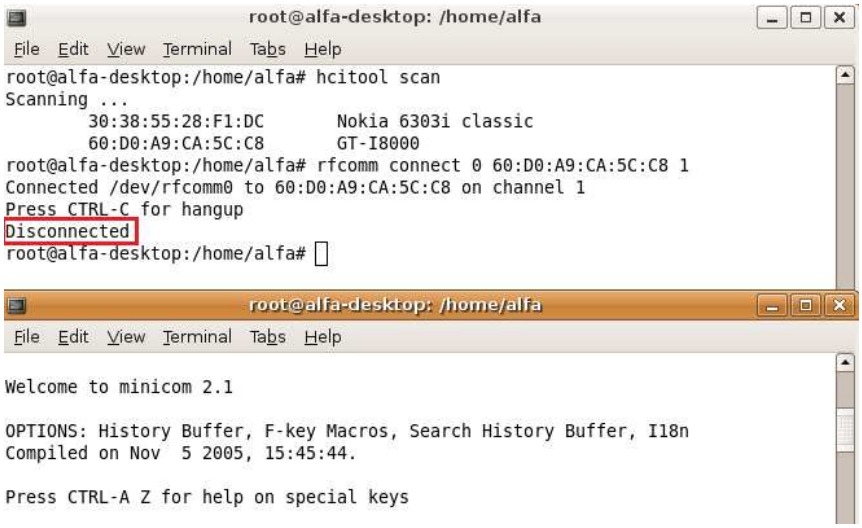



**Figura 13.** Ligação da RFCOMM desconectada assim que se liga o *minicom*.

Em relação aos outros dispositivos da Nokia, o ataque é possível. No caso do Nokia mais antigo (ano 2007), sempre que se fazia o pedido RFCOMM era solicitado a aceitação da ligação, mas com o Nokia mais recente (ano 2010) era feito um *bypass* a este pedido.

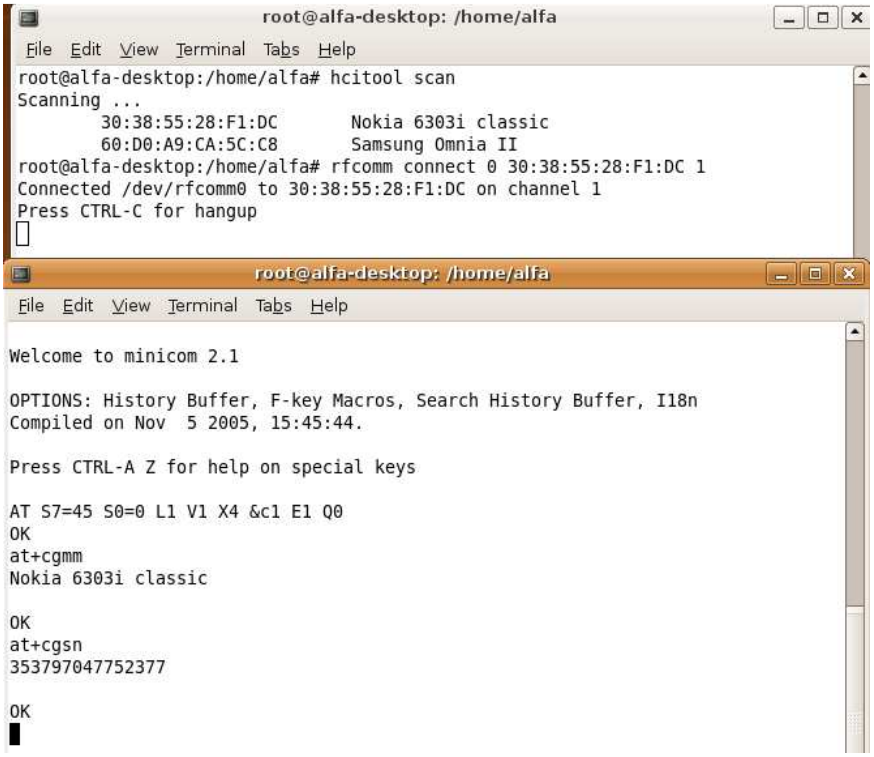



**Figura 14.** Comandos AT enviados ao Nokia do ano 2010 sem problemas.

Com estas observações verifica-se que a versão 2.1 do Bluetooth veio facilitar a interligação dos dispositivos, mas por outro lado deixa uma porta aberta para que dispositivos não seguros e que tenham sido emparelhados, possam aceder a qualquer momento ao dispositivo de Bluetooth alvo.

# 4. FIREWALL PARA BLUETOOTH

A internet é uma vasta rede que contém uma grande quantidade de informação, mas essa mesma informação pode conter conteúdo prejudicial para um computador. Estes conteúdos podem ser vírus ou *trojans* que podem apagar informações importantes, alterar informações, ou até mesmo danificar o computador. Para evitar que isto aconteça, o ideal é implementar uma barreira para bloquear esses conteúdos nocivos. Esta barreira é chamada *firewall* e existem dois tipos de *firewall*. Um é a *firewall* de *hardware* (um dispositivo físico agindo como uma barreira entre o computador e a Internet), e o segundo é uma *firewall* de *software* (uma aplicação instalada no computador ou no dispositivo móvel).

Uma *firewall* de *hardware* é um produto independente como um *router* de banda larga. Um *router* permite que computadores que estejam ligados à mesma rede possam transferir dados entre si e aceder à Internet. A *firewall* de hardware filtra os pacotes de dados, ou seja, compara o cabeçalho dos pacotes e determina os endereços IP de origem e destino. Em seguida, compara os endereços IP com as regras estabelecidas e com base nestas regras, os pacotes são transferidos ou descartados. O método de análise de dados é o *Stateful Packet Inspection* (SPI) que verifica os pacotes de dados.

Com o SPI, a *firewall* determina a origem do pacote e transfere-o, ou descarta-o, com base no pedido efetuado pelo computador. Existe um cuidado a ter em relação à *firewall*: caso dois computadores estejam ligados ao mesmo *router*, a *firewall* não verifica o conteúdo dos dados. A *firewall* assume que os dados que estão a ser transferidos são seguros e a transferência é efetuada. Portanto, se um dos computadores tiver um vírus ou um *trojan*, estes serão transmitidos aos outros computadores na mesma rede. A Figura 15 ilustra um exemplo de uma *firewall* por *hardware*.

**Figura 15.** Firewall por Hardware da marca WatchGuard, modelo NGFW XTM 800 Series (figura tirada de: [*Watchguard*]).

A principal vantagem de uma *firewall* de *hardware* é que esta protege todos os computadores de uma rede contra programas nocivos vindos da Internet. A Figura 16 ilustra um exemplo da *firewall* por hardware com a Internet e uma rede privada.


**Figura 16.** Interação da Firewall por Hardware (adaptado de [*Ricky Panchal*, 2005]).

As *firewalls* de *software* são programas que funcionam num computador. Elas monitorizam todos os portos abertos num computador e verificam toda a informação que é recebida através da Internet ou de uma rede local. O objetivo principal das *firewalls* é proteger recursos internos de ações nocivas vindas do exterior.

A *firewall* de *software* que é instalado em cada computador ou em cada dispositivo móvel é o ideal porque protege o dipositivo localmente contra atividades maliciosas tanto vindas da Internet como da rede local. No entanto, a utilização de ambas as *firewalls* (por *hardware* e *software*) em simultâneo fornecem um modo de segurança mais robusto.

## 4.1. Caraterísticas das Firewalls

As *firewalls* servem para proteger os computadores ou outros dispositivos expostos a tráfego de dados, mas podem não ser suficientes. As *firewalls* tradicionais são limitadas porque não protegem contra *malwares*, intrusões, e não filtram o conteúdo dos pacotes de dados. Devido a esta incapacidade das *firewalls* tradicionais, as *Next-Generation Firewalls* (NGFW) apareceram no mercado contemplando funcionalidades que teriam de ser adicionadas à *firewall* tradicional tornando esta mais complexa de operar e efetuar manutenção, bem como o elevado preço na aquisição das funcionalidades em falta. As seguintes características mostram o que as NGFW contêm e que não existem nas *firewalls* tradicionais [*DELL*, 2012]:

- Prevenção contra *malwares*, invasões e ataques sofisticados:
  Bloqueia vírus (programa que quando executado replica-se através da inserção de cópias de si mesmo em outros programas de

computador), *trojans* (programa que contém código malicioso que quando executado realiza determinadas ações causando a perda ou roubo de dados e possível dano do sistema), *worms* (programa autônomo que se replica a fim de se espalhar para outros computadores utilizando uma ligação de rede para se espalhar), *rootkits* (um tipo de *software* furtivo, normalmente malicioso, criado para esconder a existência de certos processos ou programas a partir de métodos normais de deteção e permitir a continuação de acesso privilegiado a um computador) e *polymorphic malware* (um código que usa um motor polimórfico de mutação, mantendo o algoritmo original intacto enquanto o próprio código muda cada vez que é executado). Esta proteção é feita na porta de entrada (*gateway*) antes que atinja a rede pessoal ou empresarial.

- Verificação de Tráfego *Secure Socket Layer* (SSL):
  SSL oferece segurança para comunicação entre dois *hosts*, sendo esta segurança baseado em integridade (medição do desempenho), autenticação (confirmação da veracidade dos dados ou do utilizador) e confidencialidade (proteção contra acesso não autorizado à informação). Geralmente é utilizado nos *browsers*, mas pode ser utilizado com qualquer protocolo que utiliza TCP como a camada de transporte.
- Controlo de Aplicações Web:
  As *firewalls* tradicionais não associam tráfego de rede com aplicações específicas. As NGFW oferecem inteligência e controle de aplicações. Isto significa que podem reconhecer o tráfego pertencente a certas aplicações e aplicam políticas de uso aceitável podendo definir largura de banda para aplicações de alta prioridade. Além disso, as NGFW permitem que os administradores de redes monitorizem e visualizem o tráfego da rede, sendo possível observar os volumes de tráfego por aplicação, a largura de banda local e determinar que tráfego fica mais lento nos horários de pico durante o dia. Esta funcionalidade de visualização de tráfego da aplicação é

uma ferramenta poderosa para solucionar problemas de capacidade no planeamento da rede. Resumindo, as NGFW conseguem bloquear aplicações que põem em perigo a segurança ou reduzir a produtividade como é o caso da partilha de ficheiros *peer-to-peer*, controlo de aplicações como é o caso de programas de mensagens instantâneas para a troca de texto (Skype e Messenger), limitar o uso de aplicações a certas horas do dia (por exemplo, Facebook e jornais online só ao meio dia e no final do horário de trabalho), e certificar que as aplicações de alta prioridade (gestão de relacionamento com clientes e processamento de pedidos) vão ter mais largura de banda do que as aplicações menos urgentes (*chat*, *vídeo streaming*, entre outros).

- Manuseamento de utilizadores e políticas de utilização:
  As *firewalls* tradicionais não conseguem ligar o tráfego de rede com os utilizadores enquanto as NGFW permitem o controlo de aplicações associados a grupos de utilizadores, ou até mesmo a um utilizador específico, permitindo aplicar políticas de uso a um nível granular como a utilização do Facebook, Twitter, LinkedIn e qualquer outro tipo de *sites* que podem ser responsáveis por centenas de horas improdutivas para os funcionários de uma empresa. No entanto, departamentos como os de *marketing* e recursos humanos podem ter acesso a estes mesmos *sites* de forma a promoverem produtos e serviços, e até mesmo encontrar candidatos para um emprego.
- Manuseamento da segurança e o desempenho da rede:
  As *firewalls* tradicionais muitas vezes forçam os administradores a optarem pela segurança em vez do desempenho da rede. Por exemplo, se os administradores da rede ativarem todas as medidas de segurança na *firewall*, o tráfego da rede pode ficar suspenso. As NGFW por outro lado têm uma maior taxa de transferência de dados devido a fatores como processadores com velocidades mais elevadas, *Central Processing Unit* (CPU) desenhados para

compreender as comunicações da rede e realizar a verificação de segurança, arquiteturas de processamento paralelo e abordagens mais eficientes de *Deep Packet Inspection* (DPI). Todas estas características fazem com que a segurança seja mais elevada e que o desempenho da rede não seja afetado [*DELL*, 2012].

Embora as *firewalls* tradicionais e as NGFW protegem a rede e o utilizador de programas nocivos vindos da Internet, todas elas seguem uma série de regras definidas pelo administrador da rede ou pelo utilizador comum. As regras da *firewall* permitem que um computador possa enviar ou receber dados de programas, serviços do sistema, computadores ou de outros utilizadores. As regras da *firewall* podem ser criadas para executar uma das três seguintes ações:

- Permitir a ligação.
- Permitir uma ligação somente se esta estiver protegida com o *Internet Protocol Security* (IPSec).
- Bloquear a ligação.

As regras podem ser criadas para tráfego de *inbound* (entrada) e para tráfego de *outbound* (saída). As regras podem ser configuradas para especificar que computadores ou utilizadores, programas, serviços ou portos e protocolos podem ter acesso. É possível também especificar qual o tipo de adaptador de rede a que a regra será aplicada, sendo estes adaptadores a LAN, Wi-Fi, acesso remoto como uma ligação de rede virtual privada (VPN), entre outros. É também possível configurar uma regra para ser aplicada à utilização de qualquer perfil ou somente a um perfil específico [*Windows Rules*, 2009].

É importante compreender a ordem pela qual as regras são aplicadas. Os seguintes passos demonstram a ordem pela qual as regras são aplicadas no Windows Server [*Windows Rules*, 2009]:

1. ***Authenticated bypass***: Estas regras contêm a opção de *Override Block Rules*, o que permite acesso de tráfego da rede que por omissão seria bloqueado. O tráfego da rede tem de ser autenticado por uma regra de segurança numa ligação separada. Estas regras permitem a administradores de sistemas terem acesso aos computadores de forma a efetuar trabalhos de manutenção ou resolução de problemas.

2. ***Block connection***: Estas regras bloqueiam todo o tráfego de entrada na rede.
3. ***Allow connection***: Estas regras verificam o tráfego de entrada na rede. Sendo o comportamento padrão de uma *firewall* bloquear o tráfego não solicitado, devem ser criadas regras que permitem que os serviços ou programas no computador que necessitam ligações externas, possam ter acesso a essas ligações e respetivo tráfego de *inbound*.
4. ***Default profile behavior***: O comportamento padrão de uma *firewall* é bloquear o tráfego de *inbound* (entrada), mas permitir todo o tráfego de *outbound* (saída). É possível alterar este comportamento nos domínios dos perfis, sendo estes perfis *Domain Profile*, *Private Profile*, e *Public Profile*. Os seguintes pontos explicam estes perfis:
    - ***Domain Profile*** (perfil de domínio): Aplicado a um adaptador de rede na qual pode ser detetado um controlador de domínio ao qual o computador está associado.
    - ***Private Profile*** (perfil privado): Aplicado a um adaptador de rede quando este estiver ligado a uma rede que é identificado pelo utilizador ou administrador como uma rede privada. A rede privada é uma rede que não está ligada diretamente à Internet, mas está por trás de algum tipo de dispositivo de segurança, como é o caso da tradução de endereços de rede ou *Network Address Translation* (NAT) do *router* ou *firewall* de *hardware*. As configurações dos perfis privados devem ser mais restritivas do que as configurações do perfil de domínio.
    - ***Public Profile*** (perfil público): Aplicado a um adaptador de rede quando este estiver ligado a uma rede pública, como é o caso dos *Hot-Spots* de Wi-Fi disponíveis em aeroportos e cafés. Quando o perfil não está definido como domínio ou privado, o perfil padrão é público. As configurações do perfil público devem ser as mais restritivas porque o computador está ligado a uma rede pública, onde a segurança não pode ser controlada. Por exemplo, um programa que aceite ligações de *inbound* vindas da Internet (como um programa para partilha de ficheiros) pode não funcionar no perfil público,

porque a configuração padrão da *firewall* irá bloquear todas as ligações de entrada para programas que não estejam na lista de programas permitidos.

Nas *firewalls* os dados são passados através dos portos (*software* específico para comunicar com o sistema operativo de um computador destino), mas uma vez que o porto é aberto (por exemplo, tráfego de HTTP porto 80, HTTPS porto 443, SMTP porto 25, UDP portos 161 e 162, e FTP portos 20 e 21), qualquer tipo de dados pode ser enviado disfarçado como tráfego seguro. As *firewalls* baseadas em portos usam os endereços IP (origem/destino) e informações das portas TCP e UDP para determinar se é permitida a passagem de pacotes entre as diferentes redes. O modo como a *firewall* filtra os dados é verificando os primeiros *bytes* do cabeçalho TCP num pacote IP, para determinar o protocolo de comunicação, por exemplo, SMTP (porto 25) e HTTP (porto 80). Por outro lado, as *firewalls* SPI tradicionais filtram o tráfego baseado em portos e protocolos. Este filtro pode bloquear ou permitir todo o tráfego no porto 80 para HTTP ou porto 443 para o tráfego HTTPS. Esta é uma abordagem que permite todo o tráfego ou nenhum. As *firewalls* mais recentes, conseguem filtrar o tráfego baseado em aplicações ou no tipo de tráfego que passa pelas portos. Por exemplo, é possível abrir o porto 80 e apenas selecionar o tráfego HTTP para aplicações específicas, *sites* ou determinados serviços [Eric Geier, 2011].

As *firewalls* são configuradas para conter uma lista de aplicações que podem aceder à Internet através de determinados portos. Portanto, se o pedido de acesso usa um porto específico, a *firewall* irá verificar o conteúdo que entra nesse porto e transmite a informação caso seja aceite. Se uma aplicação que existe num computador tentar aceder a informação existente na Internet à qual não está permitida, a *firewall* irá bloquear todas as comunicações de entrada e saída, e notifica o utilizador que o programa está a tentar aceder à Internet. Com este mecanismo o utilizador escolhe se a aplicação é segura para ter acesso à Internet ou não.

Um outro método para configurar uma *firewall* é a utilização de políticas como uma *blacklist* e uma *whitelist*. A forma como se configura uma *blacklist* e uma *whitelist* é definindo endereços de IP que são conhecidos como perigosos dentro da *blacklist* (que bloqueia todo o tipo de tráfego para estes endereços de IP), e endereços IP conhecidos como seguro dentro da *whitelist* (que só deixa passar trafego pertencente a

estes endereços de IP). Embora este método seja muito seguro, um problema surge quando se aplica uma *whitelist*. Por exemplo, ao efetuar uma pesquisa na Internet, a maioria dos *sites* que tentamos aceder serão bloqueados porque certamente não farão parte da *whitelist*. Assim sendo, o ideal para ter uma utilização normal da Internet é utilizar somente uma *blacklist* caso os endereços perigosos sejam conhecidos [*Security SW*].

Uma *firewall* pode proteger um computador e notificar o utilizador de quaisquer tentativas externas de acesso à rede ou a um computador pessoal. A *firewall* permite um enorme controlo sobre a informação que passa, e se for configurada corretamente, é muito eficaz na proteção contra programas nocivos vindos da Internet [*Ricky Panchal*, 2005].

Seja uma *firewall* pessoal que controla o fluxo de dados de um computador para outro, seja uma *firewall* de rede que controla o fluxo de dados de, e para diferentes zonas de segurança como a Internet, a LAN e a Demilitarized Zone (DMZ) que é uma sub-rede que expõe a rede de uma organização a uma rede não confiável, uma *firewall* controla que dados são permitidos e que dados não são permitidos de acordo com as regras definidas [*Abdel-Aziz, A.*, 2009].

## 4.2. Next-Generation Firewalls

As *Next-Generation Firewalls* (NGFWs) usam o *Deep Packet Inspection* (DPI) que é um filtro de pacote de dados que verifica os dados e os cabeçalhos dos pacotes. A NGFW é mais que uma *firewall* normal como foi referido acima e que apenas verifica o fluxo de dados. Esta nova *firewall* é mais completa pois integra funcionalidades de segurança que normalmente são executadas por outros *softwares*, como é o caso de *intrusion prevention* e *malware filtering*. O modo como a NGFW consegue filtrar este tipo de intrusões é analisando em profundidade o *packet payload* (informação de controlo e dados do utilizador) tomando uma decisão em aceitar ou negar o fluxo de dados. Resumindo, continua a ser um controlo de fluxo de dados mas com mais profundidade [*Abdel-Aziz, A.*, 2009]. Dada a sua flexibilidade, as NGFW podem funcionar de vários modos distintos como por exemplo:

- *Intrusion Prevention System* (IPS) – *firewall* que detecta e bloqueia as intrusões.
- *Intrusion Detection System* (IDS) – *firewall* por *hardware* que detecta intrusões e avisa o utilizador.
- *Web Application Firewall* (WAF) – *firewall* que protege as aplicações Web.

Um IPS é um tipo de *firewall* que para além de funcionar como um sistema de deteção de intrusões, permite também filtrar e bloquear dados e ligações maliciosas. Um IDS verifica o tráfego de rede e tenta mapear os dados dentro dos pacotes com uma base de dados de assinatura de ficheiros (identifica o tipo ficheiro ou o seu conteúdo) contendo informações relativas a programas nocivos, ou deteta anomalias dentro padrões de tráfego normal.

Existem algumas desvantagens na utilização de um IPS. Os IPS são desenhados para bloquear determinados tipos de tráfego que seja identificado como potencial tráfego perigoso, e não têm a capacidade de entender a lógica do protocolo de aplicações Web. Assim, os IPS não podem distinguir totalmente se um pedido é normal ou mal formado na camada de aplicação (camada *Open Systems Interconnection* (OSI) nível 7). Isto pode permitir que ataques sejam executados sem serem detetados ou prevenidos e que algum tráfego normal seja bloqueado ("falso positivo" e "falso negativo"). Os "falso positivos" é quando a *firewall* detecta uma atividade como sendo um ataque, quando na verdade não é um ataque, ou seja, quando pacotes normais são identificados como tentativas de ataque. Por outro lado, os "falsos negativos" acontecem quando a *firewall* não identifica os verdadeiros ataques.

OS IPS podem funcionar como *Host IPS* (HIPS) instalado em cada computador ou como *Network IPS* (NIPS) que monitorizam toda a rede. Os HIPS são mais granulares do que as NIPS. As HIPS monitorizam a camada de aplicação (camada OSI nível 7) de forma mais parecida com as aplicações Web. Mas HIPS ainda carecem um pouco de compreensão das linguagens de aplicações Web. Em resposta a estas deficiências, surge a *Web Application Firewall* (WAF) [*Jim McMillan*, 2009].

As WAF foram desenhadas para proteger as aplicações Web contra ataques que os IPS não conseguem impedir. As WAF podem ser aplicadas ao nível da rede ou localmente no computador. As WAF monitorizam o tráfego de, e para as aplicações.

Basicamente, a diferença entre as IPS e as WAF está na capacidade de analisar a lógica na camada 7 do modelo OSI nas aplicações Web.

Tipicamente os IPS mapeiam normalmente o tráfego contra assinaturas, enquanto as WAF verificam o comportamento ou anomalias, e a lógica dos dados que é solicitado e devolvido. As WAF estão configuradas para uma determinada aplicação em particular enquanto as outras *firewalls* protegem de forma genérica. Tipicamente as WAF protegem as *Web Applications* ou aplicações Web (aplicações por *software* executados num *web browser*) contra ataques porque conhecem a aplicação que estão a proteger. Os ataques podem ser o *SQL injection* (uma técnica de injeção de código usado para atacar aplicações orientadas a dados), *cross-site scripting* (permite que *hackers* injetem *scripts* do lado do cliente em páginas vistas por outros utilizadores), *session hijacking* (obtenção de uma sessão de computador e respetivo acesso não autorizado a informações ou serviços existentes nesse computador), *parameter or URL tampering* (manipulação de parâmetros trocados entre o cliente e o servidor a fim de modificar os dados das aplicações, tais como credenciais de utilizadores e permissões de acesso) e *buffer overflows* (violação da segurança da memória que pode ser usado para executar código ou alterar a forma como o programa funciona). Resumindo, fazem o mesmo que um IPS faz, mas por meio da análise do conteúdo de cada pacote de dados de *inbound* e de *outbound*.

As WAF são normalmente aplicadas numa espécie de *proxy* (um sistema de computador que atua como um intermediário para pedidos de clientes que buscam recursos de outros servidores) à frente das aplicações Web, para que não seja possível ver todo o tráfego na rede privada. Ao monitorizar o tráfego antes que este atinja a aplicação Web, as WAF podem analisar os pedidos antes de os passar, sendo uma vantagem em relação às IPS, isto porque, tendo as IPS sido desenhadas para verificar todo o tráfego da rede, as IPS não conseguem analisar a camada de aplicação Web tão bem como as WAF.

As WAF não só detetam ataques que são conhecidos, como também detetam (e podem impedir) novos tipos de ataques que sejam desconhecidos ao observar padrões incomuns ou inesperadas no tráfego. Por exemplo, se uma WAF detetar que a aplicação está a retornar mais dados do que o esperado, o WAF pode bloquear esta aplicação e alertar o utilizador [*Jim McMillan*, 2009].

A WAF é fundamental num sistema de segurança *Defense-in-Depth* porque consegue bloquear o ataque antes que este possa ser executado. A WAF filtra os dados de entrada e saída entre as várias aplicações. Esta *firewall* pode ser uma das NGFW usando o DPI e o *intrusion prevention* no *core* do seu funcionamento. A WAF proporciona também a possibilidade de ser modificada e aperfeiçoada [*José Fonseca*, 2011].

O *Defense-in-Depth* é baseado no princípio que a segurança é melhorada se existir redundância de sistemas de segurança e que haja várias camadas de mecanismos de segurança como o IDS, IPS, WAF, antivírus, *antispyware*, *antisspam*, entre outros, tanto ao nível da rede, do sistema operativo e ao nível das aplicações [*NSA DEFENSE IN Depth]*.

O *Defense-in-Depth* é um conceito de defesa por camadas, desde a segurança por parte do utilizador onde responsabilidades são atribuídas, passando pela tecnologia em que esta terá de ser suficientemente robusta de modo a detetar intrusões, e por fim, as operações que são atividades que sustentam a segurança de uma organização, como por exemplo, manutenção do sistema de segurança com atualizações de antivírus, controlo da lista de acessos, *security patches* (atualizações de segurança), monitorizar e atuar contra intrusões ou ameaças, entre outos. A Figura 17 ilustra o conceito de *Defense-in-Depth*.


**Figura 17.** Defesa Por Camadas (figura tirada de: [*Defense in Depth*, NSA]).

## 4.3. Proposta de uma Firewall para Bluetooth

Após análise dos diversos tipos de *firewall*, dado os problemas encontrados, conclui-se que o Bluetooth necessita de uma *firewall* baseado numa WAF. O filtro das ligações de Bluetooth tem de ser verificado antes que qualquer ligação seja realizada, sendo a segurança aplicada antes da conectividade com qualquer aplicação ou

procedimento de emparelhamento do Bluetooth. Dadas as especificidades da mobilidade e do pequeno tamanho dos dispositivos, o tipo de *firewall* terá de ser por *software* e instalado no próprio dispositivo móvel (como qualquer *software*). Esta *firewall* pode ser personalizada, o que permite algum controlo sobre as funcionalidades de proteção. A personalização da *firewall* irá proporcionar ao utilizador a opção de escolher um perfil para a ligação do Bluetooth, tal como uma WAF que está adaptada para proteção ao nível de cada aplicação, bem como a opção de poder inserir determinados dispositivos Bluetooth dentro de uma *blacklist.*

### 4.3.1. Análise para Implementação da Firewall

No Capítulo 3 verificou-se que todos os ataques passavam pelo protocolo RFCOMM. Sendo este protocolo a entrada de ligações indesejadas, o objetivo passa por proteger o dispositivo Bluetooth ao nível deste protocolo. Assim sendo, a *firewall* irá filtrar todas as comunicações que utilizam o protocolo RFCOMM e terá capacidade de filtrar todos os ataques relativos ao OBEX, WAP, UDP e TCP.

A Figura 18 ilustra a implementação da *firewall* no *Bluetooth protocol stack* [O Anexo A contém o artigo científico - *João Alfaiate et al.*, 2012].

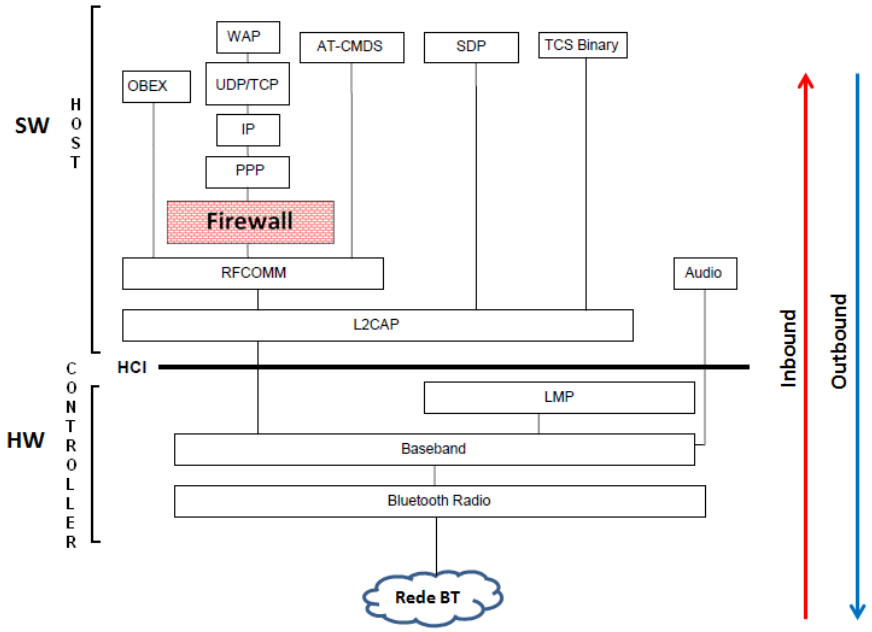



**Figura 18.** Bluetooth Protocol Stack com a Firewall (adaptado de [*João Alfaiate et al.*, 2012]).

Observando a Figura 18, verifica-se que o *Bluetooth Protocol Stack* é formado por diversas camadas de *hardware* e *software*. A interligação entre a camada de *hardware* (*controller* - módulo de rádio) com a camada do *software* (*host* - protocolos de comunicação) é efetuada pelo *Host Controller Interface* (HCI) que é a camada de transporte dos dados. A *firewall* de Bluetooth funcionará logo a seguir ao protocolo de RFCOMM, filtrando toda a informação antes que esta passe para os protocolos acima do RFCOMM.

O fluxograma ilustrado na Figura 19 mostra como a *firewall* interage perante as ligações de Bluetooth e mostra também como estas são tratadas caso sejam novas ligações ou ligações previamente configuradas.

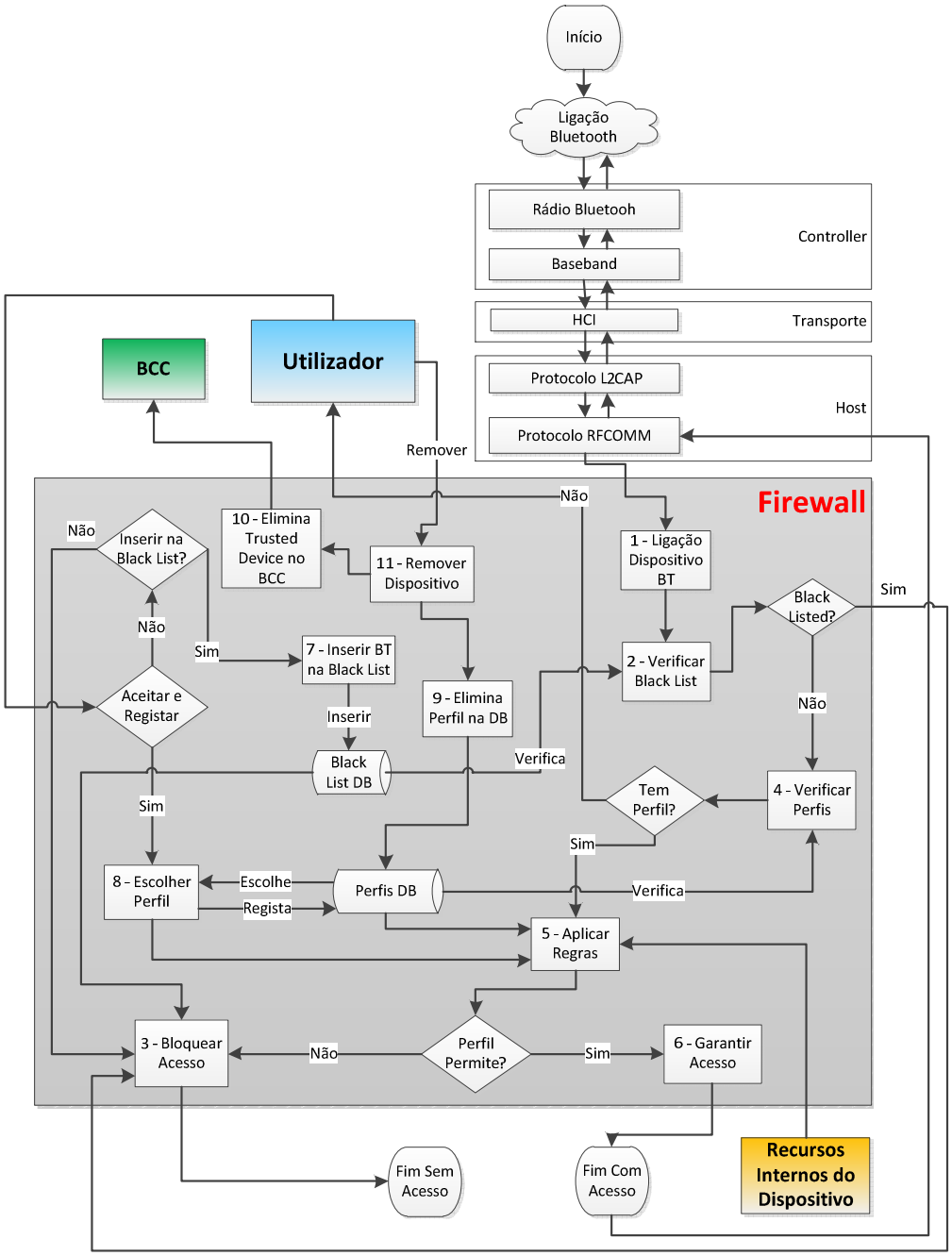



**Figura 19.** Fluxograma com a Firewall.

Sendo o Bluetooth uma tecnologia de redes sem fios, todos os dispositivos contêm um módulo de rádio que é composto pelo rádio do Bluetooth, que é o módulo de Rádio Frequência (RF) que recebe e transmite os dados, e o *baseband*, que é um controlador de ligação que estabelece e administra a ligação de RF entre os dispositivos

de Bluetooth. Quando um dispositivo Bluetooth estabelecer uma ligação com o módulo de rádio, a informação é passada ao HCI que é responsável por transmitir os dados do módulo de rádio para os protocolos de comunicação. O protocolo que recebe a informação vinda do HCI é o L2CAP. O L2CAP é responsável por passar os pacotes de dados do HCI para as camadas de protocolos acima tendo em conta o seguinte:

- Multiplexagem dos dados consoante o protocolo da camada superior.
- Segmentação (e vice-versa) dos pacotes de dados.
- Gestão na transmissão de dados a um grupo de outros dispositivos Bluetooth.
- Qualidade de Serviço (QoS) na gestão dos protocolos das camadas superiores.

Após o L2CAP filtrar os dados e passar os mesmos para o protocolo RFCOMM (1 – Ligação Dispositivo BT), a *firewall* irá verificar se o dispositivo Bluetooth que se está a ligar pertence à *blacklist* (2 – Verificar Black List). Caso o dispositivo pertença à *blackist*, a ligação será imediatamente bloqueda (3 – Bloquear Acesso). Se o dispositivo não pertence à *blackist*, a *firewall* irá verificar se o dispositivo já tem algum perfil atribuído (4 – Verificar Perfis). Caso tenha um perfil atribuído, serão aplicadas as regras correspondentes ao perfil atribuído (5 – Aplicar Regras). Neste mesmo passo (n.º5), a *firewall* irá interagir com os recursos internos do dispositivo móvel (contactos, calendário, *email*, fotografias, entre outros), para que esta informação esteja disponível ao respectivo perfil. De salientar que a *firewall* será responsável por garantir que recursos internos do dispositivo móvel que não pertençam a um determinado perfil, não possam de modo algum estar disponíveis a esse perfil ou que haja uma fuga desta informação para um perfil não autorizado. Esta é uma parte muito sensível porque irá requerer APIs específicas para cada sistema operativo dos dispositivos móveis. Após as regras serem aplicadas, a *firewall* filtra que tipos de acessos o dispositivo Bluetooth externo poderá ter, bloqueando os acessos a que não tem direito (3 – Bloquear Acesso) e permitindo acesso à informação a que tem direito (6 – Garantir Acesso). No caso da ligação de Bluetooth ser uma nova ligação e não ter um perfil atribuído, será solicitado ao utilizador para aceitar ou negar o acesso. Caso o utilizador negar o acesso, será sugerido para inserir o dispositivo na *blacklist*. Se o utilizador aceitar colocar o dispositivo na *blacklist*, a base de dados da *blacklist* será

actualizada com a informação desse dispositivo (7 – Inserir BT na Black List), e a ligação será automaticamente bloqueada como ilustrado com a linha a desde a *Black List DB* até ao processo que bloqueia o acesso (3 – Bloquear Acesso). Se por algum motivo em particular o utilizador não pretender incluir o dispositivo na *blacklist*, a ligação é bloqueada sem outro tipo de intervenção, podendo este dispositivo de Bluetooth no futuro efectuar outro pedido de acesso. Em caso do utilizador pretender aceitar e registar o acesso do dispositivo de Bluetooth, será pedido ao utilizador para escolher um perfil (8 – Escolher Perfil). Por omissão a *firewall* contemplará por defeito três perfis (*@Home, Temporary, E-Commerce*). O utilizador não terá de configurar regras visto que para estes três perfis estarão definidas, entretanto o utilizador poderá definir novos perfis definindo as regras para estes. O perfil *Temporary* tem a particularidade de conter um temporizador que serve para remover automaticamente um dispositivo de Bluetooth após um determinado período de tempo (9 – Elimina Perfil na DB). Mas a remoção do dispositivo na base de dados dos perfis não é suficiente, isto porque, por cada ligação Bluetooth aceite, o Bluetooth por si regista a informação do dispositivo no BCC, que é um mecanismo que regista os dispositivos Bluetooth externos como *trsuted devices*, permitindo acesso sem novo emparelhamento aos dispositivos que estão listados dentro da base de dados do BCC. Portanto, ao ser removido o dispositivo da base de dados dos perfis, terá de haver um método que remova este mesmo dispositivo da base de dados do BCC (10 – Elimina Trusted Device no BCC). O mesmo se aplica quando o utilizador decidir remover um dispositivo qualquer das suas ligações, ou seja, quando se remove da base de dados dos perfis, o BCC será automaticamente actualizado com a eliminação deste mesmo registo (11 – Remover Dispositivo). Por fim, após o utilizador definir a que perfil o dispositivo será inserido, serão aplicadas as regras e os processos mencionados nas caixas de processos n.º 3, 5 e 6.

A *firewall* Bluetooth tem a particularidade de ser parecido com uma WAF porque filtra o dispositivo Bluetooth que se tenta ligar, e analisa se este se pode ligar ou não, e executa uma série de procedimentos. Este método é como a WAF que filtra as ligações ao nível da aplicação e não permite que a ligação seja estabelecida antes de esta ser confirmada.

### 4.3.2. Atribuição de Perfis

A *firewall* além de filtrar as ligações de forma a evitar que ligações indesejadas sejam efetuadas, irá associar todas as ligações seguras a um tipo de perfil, sendo que as definições de segurança variam de acordo com o perfil de modo a que haja mais liberdade na utilização.

A utilização dos perfis do Bluetooth varia consoante a necessidade do utilizador e também do tipo de dispositivo móvel que é utilizado. Tomando como exemplo a gama de dispositivos móveis que a Apple oferece, podemos verificar através da Figura 20 que os telemóveis estão mais vulneráveis aos ataques do que outros dispositivos móveis (iPod e iPad) visto terem mais perfis de Bluetooth associados.

| Device | Hands-Free Profile (HFP 1.6) | Phone Book Access Profile (PBAP) | Advanced Audio Distribution Profile (A2DP) | Audio/Video Remote Control Profile (AVRCP 1.4) | Personal Area Network Profile (PAN) | Human Interface Device Profile (HID) | Message Access Profile (MAP) |
|---|---|---|---|---|---|---|---|
| iPhone 4 and later | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| iPhone 3GS | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | – |
| iPhone 3G | ✓ | ✓ | ✓ | ✓ | ✓ | – | – |
| Original iPhone | ✓ | ✓ | – | – | – | – | – |
| iPad 2 and later | ✓ | – | ✓ | ✓ | ✓ | ✓ | – |
| iPad (1st generation) | – | – | ✓ | ✓ | ✓ | ✓ | – |
| iPod touch (4th generation and later) | ✓ | – | ✓ | ✓ | ✓ | ✓ | – |
| iPod touch (2nd and 3rd generation) | – | – | ✓ | ✓ | ✓ | ✓ | – |
| **Total #** | **6** | **4** | **7** | **7** | **7** | **6** | **1** |

**Figura 20.** Apple e Bluetooth [*iOS Bluetooth profiles*, 2013].

De modo a saber quais os dispositivos de Bluetooth mais utilizados, é necessário saber o número de dispositivos que existem para os diferentes objetos e modelos que as marcas comercializam. A Figura 21 mostra a lista do número de dispositivos Bluetooth por área e que estão aprovados pelo *Special Interest Group* (SIG)

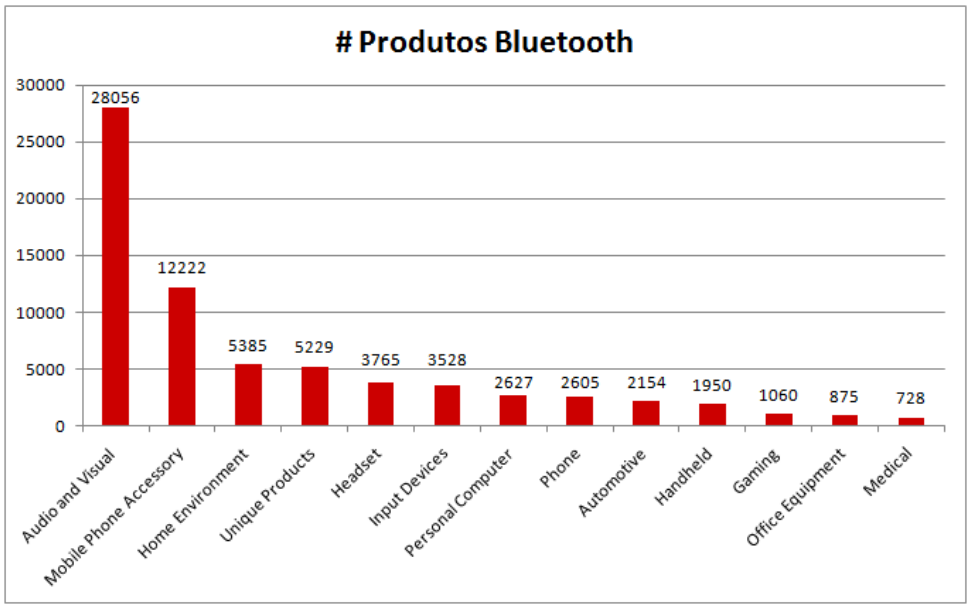



**Figura 21.** Número de Dispositivos Bluetooth [*Bluetooth Product Directory*].

Tendo em conta os dados mostrados na Figura 21, conclui-se que o Bluetooth é mais utilizado para áudio e vídeo e de seguida para acessórios de telemóveis. Os equipamentos utilizados em casa surgem na terceira posição.

Por omissão, a *firewall* virá com três perfis que corresponderá às necessidades de um utilizador comum. No entanto, o objetivo é no futuro proporcionar ao utilizador a possibilidade de poder criar os seus próprios perfis de acordo com a utilização pretendida. Estes perfis podem por exemplo contemplar jogos, automóveis, e rede sociais (*social networking*). No entanto os perfis por omissão conseguem de uma forma geral proteger o utilizador contra a maioria dos cenários que poderão ter impacto em termos de segurança. Os perfis que a *firewall* disponibilizará são *@Home*, *Temporary* e *E-Commerce*.

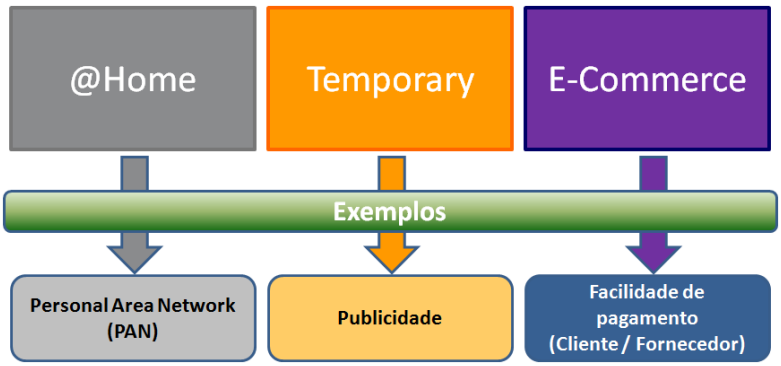



**Figura 22.** Perfis de Utilizador.

Perfis:

- *@Home* – Este perfil é intencionado para o uso de dispositivos Bluetooth do dia-a-dia, como por exemplo, um auricular, uma PAN ou rede doméstica, entre outros.
- *Temporary* – Este perfil serve para emparelhar dispositivos Bluetooth em que não se sabe qual a origem do dispositivo a ser adicionado, como por exemplo, publicidade, dispositivos de terceiros para transferência de dados, entre outros.
- *E-Commerce* – Este perfil serve para compras ou pagamentos utilizando dispositivos de fornecedor de serviços.

Quando um novo dispositivo Bluetooth tenta ligar-se, a *firewall* irá pedir ao utilizador para aceitar ou negar o acesso, junto com a possibilidade de associar a ligação ao perfil desejado. Ao separar as diversas ligações de Bluetooth do telemóvel por perfis, permite-se que o utilizador tenha controlo sobre estas mesmas ligações e acima de tudo, que fique despreocupado em relação a certas ligações que possam existir. Por exemplo, ao ligar-se a um dispositivo de Bluetooth publicitário, o utilizador escolherá o perfil de *Temporary* onde esta ligação não poderá de modo algum transmitir dados para fora do dispositivo móvel, e ainda poderá definir com este perfil um temporizador que após um determinado número de horas ou dias, esta ligação seja automaticamente removida sem necessitar intervenção do utilizador.

#### 4.3.2.1. Perfil @Home

Hoje em dia a maioria dos equipamentos já incorporam um dispositivo Bluetooth para comunicação. Estes equipamentos por norma são de uso doméstico onde o utilizador cria uma *Personal Area Network* (PAN) para facilitar a interligação.

A forma como este perfil atua é do seguinte modo: nenhum dispositivo Bluetooth poderá aceder à PAN se não pertencer ao perfil *@Home*, ou seja, todas as tentativas de outras ligações em aceder a qualquer dispositivo da PAN será automaticamente bloqueada. A Figura 23 ilustra este cenário.

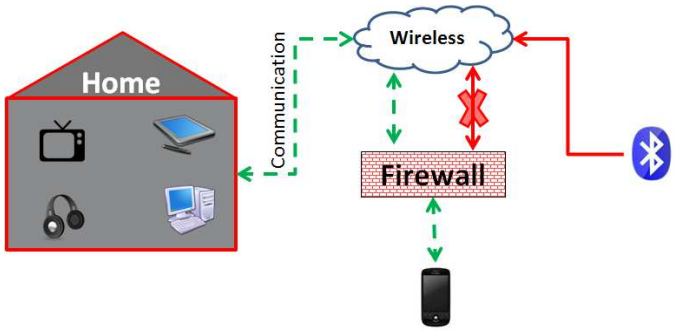



**Figura 23.** Perfil @Home.

A identificação do dispositivo nesta primeira fase para o perfil *@Home* será com o UUID e com o MAC *address*.

#### 4.3.2.2. Perfil Temporary

Como já foi descrito anteriormente, o perfil *Temporary* tem como objetivo servir como uma caixa fechada que apenas permite receber informação (*inbound*) proibindo qualquer comunicação para o exterior (*outbound*). Este perfil é muito útil quando queremos nos ligar a dispositivos que não conhecemos antemão por algum motivo em particular.

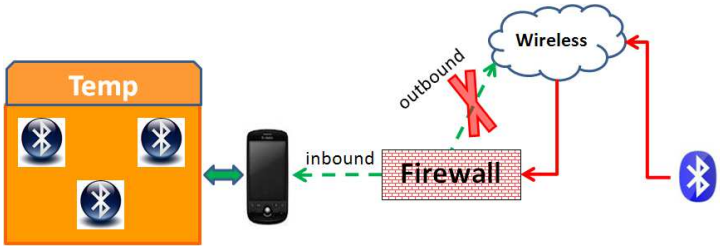



**Figura 24.** Perfil Temporary.

O registo do dispositivo será apenas com MAC *address*, mas será aplicado também a regra que remove automaticamente os dispositivos após um determinado período de tempo.

#### 4.3.2.3. Perfil E-Commerce

Uma das soluções iniciais para pagamentos utilizando uma rede sem fios foi o Bluetooth. Esta tecnologia nunca vingou no mercado como opção de pagamento talvez devido à desconfiança da sua segurança. De facto, como foi descrito, o Bluetooth tem tido problemas de segurança que ainda não foram totalmente resolvidas.

Ao criar uma *firewall* especificamente para Bluetooth e acima de tudo funcionando no telemóvel do utilizador, pode potenciar a utilização desta tecnologia para pagamentos que utilize uma rede sem fios, tornado mais prático tanto para o consumidor bem como para o fornecedor do serviço. Mas a segurança não pode ficar só pela parte do utilizador, terá de haver um encadeamento de camadas de segurança como o *Defense-in-Depth* [*Defense in Depth*, NSA].

Neste perfil de *E-Commerce*, a *firewall* não será suficiente para garantir por si só segurança total, porque não se consegue prever o que poderá passar entre o dispositivo do consumidor e o dispositivo do fornecedor. A solução encontrada é a elaboração de um *software* que terá de funcionar do lado do fornecedor de serviços. Chaves de segurança criadas no momento para o pagamento serão utilizadas para garantir que o dispositivo que pretende pagar o serviço é de facto o dispositivo em questão. O conceito de *Defense-in-Depth* terá de ser aplicado para que a segurança seja aplicada desde o início da transação até ao seu término. Terá ainda de haver algum mecanismo de autenticação e encriptação dos dados, o que é obrigatório por lei e regulamentos cooperativos referentes às transações eletrónicas (por exemplo, o *Payment Card Industry* (PCI) *Data Security Standard* (DSS) que são *standards* internacionais desenvolvidos para incentivar e aperfeiçoar a segurança de dados em pagamentos eletrónicos por cartão [*PCI DSS*, 2010]). A Figura 25 ilustra o cenário de uma forma muito simples.

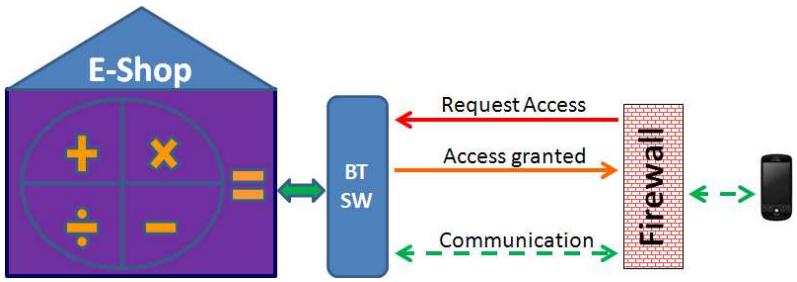



**Figura 25.** Perfil E-Commerce.

#### 4.3.2.4. Regras dos Perfis da Firewall Bluetooth

A associação das ligações aos perfis tem de seguir regras, que no seu conjunto diferem em cada perfil. Novas regras podem ser acrescentadas posteriormente. Nesta fase inicial as regras propostas são mostradas na Tabela 11, que também já inclui

mais três perfis como exemplo (Jogos, Automóvel e Redes Sociais). As regras por omissão (*@Home*, *Temporary* e *E-Commerce*) estão salientadas com a cor cinzenta.

**Tabela 11.** Regras dos Perfis de Bluetooth

| | **PERFIS** | | | | | |
|---|---|---|---|---|---|---|
| **REGRAS** | **@Home** | **Temporary** | **E-Commerce** | **Jogos** | **Automóvel** | **Redes Sociais** |
| Guardar Emparelhamento | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Autenticação UUID | ✓ | | ✓ | ✓ | ✓ | ✓ |
| Autenticação MAC | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Tráfego *Inbound* | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Tráfego *Outbound* | ✓ | | ✓ | ✓ | ✓ | ✓ |
| *Stealth Mode* | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Remoção Automática do Dispositivo BT | | ✓ | | ✓ | | ✓ |
| Acesso aos Contactos | ✓ | | | | ✓ | |
| Acesso ao Calendário | ✓ | | | | ✓ | |
| Fotografias | ✓ | | | | | ✓ |
| Email | ✓ | | | | ✓ | |
| Dados do GPS | ✓ | | | | ✓ | ✓ |

Após os dispositivos estarem associados a um perfil, sempre que estes dispositivos tentem ligar-se, a *firewall* irá filtrar endereços MAC e o UUID como mostrado na Tabela 11. Um dispositivo Bluetooth não pode pertencer a mais de que um perfil porque as regras aplicadas ao dispositivo poderão entrar em conflito e resultar em falhas na sua utilização. A *firewall* também será responsável por monitorizar o tráfego e alertar o utilizador em caso de ações suspeitas. A Tabela 11 mostra também a opção de

*Stealth Mode* que é um mecanismo que ajuda a impedir que utilizadores mal-intencionados descubram informações sobre os dispositivos móveis e serviços que são executados.

A capacidade da *firewall* de Bluetooth para autenticar ligações por perfis de utilizador é uma nova abordagem para proteger os utilizadores de problemas de segurança devido ao Bluetooth. Outras abordagens concentram-se em problemas específicos, como a proteção contra a propagação de *malware* utilizando o *Blue-Watchdog* [*Mohamed GHALLALI et al.*, 2011] ou no melhoramento na encriptação de comunicação [*Yu Xin et al.*, 2009]. Estas abordagens são no entanto limitadas para o que se propõe porque estas abordagens protegem contra *malwares* e melhoram a encriptação de comunicação, mas não fornecem uma proteção contra a globalidade de ataques que existem ou que possam surgir, não fornecendo assim a proteção holística que uma *firewall* de Bluetooth é capaz.

## 4.4. Plataforma para a Implementação da Firewall

O desenvolvimento de um *software* requer algum estudo prévio para que tenha sucesso, e o sucesso tem a ver com a expansão e crescimento do produto. De forma a ter maior relevância, a *firewall* deverá ser desenvolvida para o tipo de dispositivos móveis com mais penetração no mercado. Após uma pesquisa sobre qual o sistema operativo mais utilizado, verificou-se que o Android lidera o mercado com um volume de vendas em 2013 de 793,6 milhões de unidades, enquanto o iOS obteve um valor bastante abaixo fixando-se nos 153,4 milhões de unidades. Se observarmos a tendência dos utilizadores, o Android continua a ganhar terreno em relação ao iOS com um aumento de vendas de 36,98% de 2012 para 2013, enquanto o iOS cresceu 11,40%. A Figura 26 mostra o ranking dos sistemas operativos dos dispositivos móveis mais vendidos em 2012 e 2013.

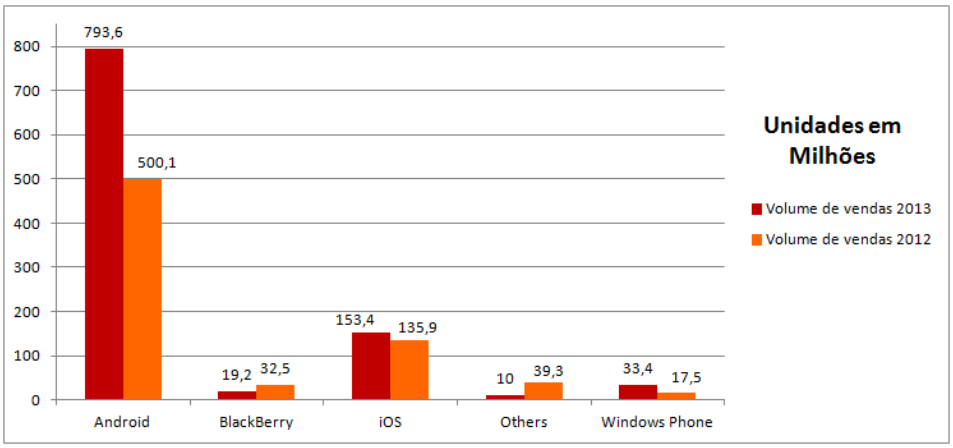



**Figura 26.** Volume de dispositivos móveis vendidos em 2012 e 2013 [*IDC - Smartphone OS*, 2014]

Tendo em conta que o Android é um líder destacado nas vendas dos telemóveis, a plataforma mais adequada para o desenvolvimento do *software* e que abrangerá a maioria dos telemóveis será o Java. A plataforma Java fornece uma série de APIs para o desenvolvimento de aplicações para Bluetooth. O *Java 2 Micro Edition* (J2ME) é focado no desenvolvimento de *software* para pequenos dispositivos com limitação de memória e de processamento tais como telemóveis, *smartphones*, PDA, *tablets*, entre outros.

Esta secção descreve as classes, os objetos e os parâmetros que são fundamentais para a elaboração da *firewall* utilizando o J2ME. As classes e parâmetros são focados no desenvolvimento de *software* para dispositivos sem fios, e acima de tudo, tendo em conta a segurança como é o caso da autenticação, encriptação e autorização.

### 4.4.1. Plataforma J2ME

Utilizando o J2ME como a plataforma de desenvolvimento, há que ter atenção que nem todos os protocolos existentes no Bluetooth são suportados pela especificação do *Java APIs for Bluetooth Wireless Technology* (JABWT). O JABWT só fornece um conjunto *standard* de APIs para o desenvolvimento do *software* [*JSR-82*, 2002]. O objetivo do JABWT é minimizar o número de classes, permitindo assim uma lógica de simples de aprender e programar. No total existem 21 classes. O JABWT tem como alvo dispositivos com processamento e memória limitadas, e cuja energia é

fornecida principalmente por bateria, o que vai ao encontro do pretendido para a *firewall.*

Como o Bluetooth *protocol stack* é por sua vez constituído por diversos protocolos, a Tabela 12 mostra quais os protocolos suportados pela especificação do JABWT.

**Tabela 12.** Protocolos suportados pelo JABWT [*JSR-82*, 2002]

| Protocolos | JSR82 - API |
|---|---|
| L2CAP | Suportado |
| RFOMM | Suportado |
| OBEX | Suportado |
| SDP | Suportado |
| LMP | Suportado |
| Audio | Não suportado |
| TCS | Não suportado |
| BNEP | Não suportado |

### 4.4.2. Análise de APIs

As APIs são um conjunto de rotinas existentes numa plataforma de programação e que auxiliam o programador fornecendo funcionalidades que poderão ser implementadas no *software* a ser desenvolvido. A utilização de APIs evita que o programador crie funções desde raiz, que de certa forma acelera o desenvolvimento. No caso do J2ME, irá ser analisado as APIs relacionadas com protocolos de comunicação de redes sem fio, e acima de tudo APIs relacionadas com segurança.

As classes e interfaces utilizadas para a comunicação são fornecidas pelo *Generic Connection Framework* (GCF). A comunicação do RFCOMM começa com o GCF, sendo que uma *string* de ligação é enviada para o `Connector.open()` de forma a estabelecer a ligação. Para ligações de clientes, o objeto `StreamConnection` é retornado pelo `Connector.open()`. O `Connector.open()`retorna o objeto

`StreamConnectionNotifier` se uma ligação de servidor for utilizada. Uma vez que a ligação entre cliente e servidor é estabelecida, a comunicação entre o cliente e servidor é efetuada por via de `InputStream` e `OutputStream`. A Figura 27 mostra estas ligações.

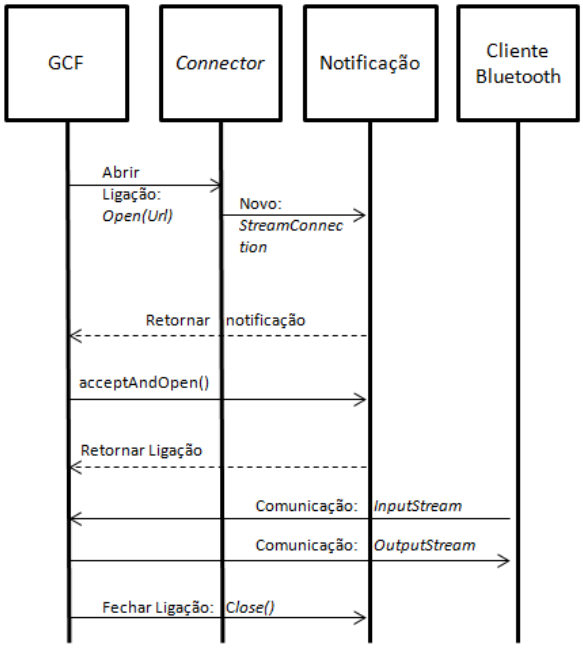



**Figura 27.** GCF e a ligação cliente.

#### 4.4.2.1. Servidor como Firewall

Todas as comunicações por RFCOMM começam por `Connector.open()` e com uma *string* de ligação válida. As *strings* de ligação passadas ao `Connector.open()` têm a seguinte estrutura:

| [Protocolo identificador]: | [UUID] | [parâmetros] |
|---|---|---|
| `btspp://localhost:` | `102030405060708090A1B1C1D1D1E100;` | `name=SPPEx` |

Em RFCOMM, tanto para o servidor como para o cliente, o protocolo é sempre o *Bluetooth Serial Port Profile* (BTSPP) enquanto o *Universally Unique Identifier* (UUID) e os parâmetros dependem se a ligação é de cliente ou de servidor. Em relação aos parâmetros, não é obrigatório incluir, mas caso seja pretendido, a Tabela 13 mostra os parâmetros que podem ser utilizados bem como as respetivas opções.

**Tabela 13.** Parâmetros e respetiva implementação

| Parâmetro | Opção | Implementação |
|---|---|---|
| `name` | Qualquer string válida | Servidor |
| `authenticate` | `true` ou `false` | Cliente e servidor |
| `encrypt` | `true` ou `false` | Cliente e servidor |
| `authorize` | `true` ou `false` | Servidor |
| `master` | `true` ou `false` | Cliente e servidor |

Após o `Connector.open()` retornar o `StreamConnectionNotifier`, a ligação está pronta a ser executada. O método `acceptAndOpen()` deve ser chamado depois do `Connector.open()`, mas antes de chamar o `acceptAndOpen()`, a *firewall* deverá ter um procedimento de alerta que poderá ser com um *try* e *catch* ou outro tipo de código, por forma a poder controlar todas as ligações mesmo sendo estes um *trusted device*. Uma das falhas detetadas nos testes de ataque é que alguns *trusted device* aceitavam a ligação sem alertar o utilizador enquanto outros avisavam e só procediam com a ligação caso o utilizador aceitasse. O ideal é a *firewall* filtrar qualquer tentativa de ligação ao telemóvel. De salientar que um dispositivo Bluetooth pode ter várias ligações RFCOMM em simultâneo, pelo que será prudente analisar todas as tentativas de ligação ao telemóvel que utilize o protocolo de RFCOMM, mesmo que a primeira ligação seja aceite. A Figura 28 mostra estas ligações.

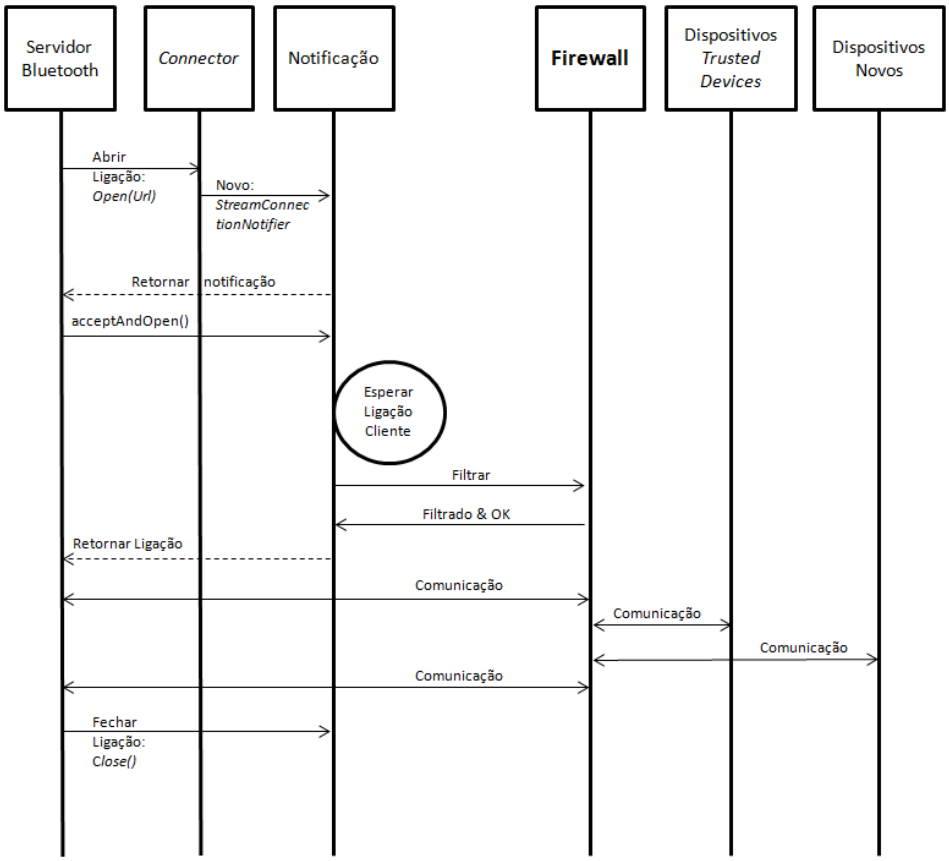



**Figura 28.** Ligação do servidor Bluetooth e filtro da Firewall com outros dispositivos Bluetooth.

Embora o Bluetooth tenha procedimentos para emparelhar dispositivos (*pair devices*), a *firewall* terá de ter a responsabilidade de filtrar todas as ligações. Assim sendo, as novas ligações a serem feitas, bem como as que já foram aceites e guardadas como fidedignas, mesmo as que já estejam emparelhadas, serão filtradas. Assim certifica-se que todas as ligações serão analisadas para que haja mais segurança.

Ao efetuar uma ligação física ou sessão entre dispositivos, o método `connector.open` é utilizado. No J2ME existem quatro tipos de invocações para este método, sendo cada um específico para determinada comunicação. Estas invocações são [*developerWorks*]:

- Invocação de comunicação por *http*:

```
Connection conn = Connector.open("http://www.google.com");
```

- Invocação de comunicação por *socket* (*stream* de dados):

```
Connection conn = Connector.open("socket://localhost:9000");
```

- Invocação de comunicação por *socket* (*datagram* – pacote de dados):

```
Connection conn = Connector.open("datagram://:9000");
```

- Invocação de comunicação por uma porta série:

```
Connection conn = Connector.open("comm:0;baudrate=9000");
```

- Invocação de comunicação por ficheiro *I/O*:

```
Connection conn = Connector.open("file://myfile.dat");
```

No caso específico do Bluetooth, é utilizado o protocolo de RFCOMM. A comunicação invocada para este estudo é o:

```
Connection conn = Connector.open("socket://localhost:9000");
```

Neste método uma *string* de ligação é enviado para o `connector.open` por forma a estabelecer a ligação.

O Anexo D contém um exemplo de código fonte que mostra como as ligações de RFCOMM são aceites e um método para visualizar as mensagens do cliente e a *string* utilizada.

#### 4.4.2.2. Segurança Existente

Antes de iniciar qualquer desenvolvimento de *software* convém saber antemão as funcionalidades que a especificação oferece de forma a não investir tempo no desenvolvimento e chegar depois à conclusão que as funcionalidades existentes não são suficientes para o pretendido. Esta secção mostra os métodos e as classes de segurança existentes tais como a autenticação e encriptação, e que são fundamentais para a *firewall*.

O método *open* da classe `javax.microedition.io.Connector` é utilizado para o início da ligação entre dispositivos. Este método é utilizado pelo servidor por forma a esperar que um cliente se ligue ao dispositivo [*JSR-82*, 2002].

Sendo que numa ligação de Bluetooth existe um servidor e um cliente, as aplicações de servidor ou de cliente podem acrescentar parâmetros nos argumentos da *string* da ligação do `connector.open`. Isto faz com que diferentes ligações que envolvam serviços diferentes possam ter níveis de segurança também diferentes, e estes argumentos terão de ser definidos consoante a necessidade.

Nem todos os sistemas de Bluetooth suportam a autenticação [*JSR-82*, 2002] e mesmo suportando, o simples facto de ter a autenticação como `authenticate=true` pode entrar em conflito com as configurações de segurança do dispositivo que o utilizador tenha estabelecido através do BCC. Por exemplo, caso a

autenticação não seja suportada e o método `authenticate=true` tenha sido utilizado, a exceção `BluetoothConnectionException` é lançado contra o `connector.open()`.

A encriptação é fundamental na comunicação por forma a evitar que a informação seja facilmente obtida através de *Eavesdropping*. O parâmetro de `encrypt` pode ser utilizado de duas formas:

- `encrypt=true`: a implementação encripta todas as comunicações.
- `encrypt=false`: encriptação não é exigida pela aplicação do servidor, mas poderá ser utilizado caso a aplicação do cliente necessite deste parâmetro.

Caso o parâmetro de `encrypt` não esteja presente na *string* de ligação, significa o mesmo que ter `encrypt=false`.

Ao conjugar a autenticação e encriptação, só algumas combinações são possíveis. A Tabela 14 mostra estas combinações.

**Tabela 14.** Parâmetro *Authenticate* vs Parâmetro *Encrypt*

| Parâmetro *authenticate* | Parâmetro *encrypt* | Combinação |
|---|---|---|
| `True` | `true` | Válida |
| `True` | `false` | Válida |
| `false` | `false` | Válida |
| `false` | `true` | Inválida |
| ----- | `true` | Autenticação ausente equivale a autenticação `true`. |

A autorização serve para que um utilizador de um dispositivo servidor possa conceder ou não acesso a um determinado serviço a um dispositivo cliente. Estes pedidos de autorização podem depender na aceitação do utilizador do servidor por cada vez que se executa este pedido, ou pode depender na consulta da lista de *trusted devices*, tendo neste caso acesso a todos os serviços.

O parâmetro de `authorize` pode ser utlizado de duas formas:

- `authorize=true`: A implementação consulta o BCC por forma a saber se o cliente tem ou não acesso ao serviço solicitado.
- `authorize=false`: Todos os clientes têm acesso ao serviço solicitado.

Caso o parâmetro de `authorize` não esteja presente na *string* de ligação, significa o mesmo que ter `authorize=false`.

Ao conjugar a autenticação e a autorização só algumas combinações são possíveis. A Tabela 15 mostra estas combinações.

**Tabela 15.** Parâmetro *Authenticate* vs Parâmetro *Authorize*

| **Parâmetro *authenticate*** | **Parâmetro *authorize*** | **Combinação** |
|---|---|---|
| `true` | `true` | Válida |
| `true` | `false` | Válida |
| `false` | `false` | Válida |
| `false` | `true` | Inválida |
| ----- | `true` | Autenticação ausente, equivale a autenticação `true`. |

Além da classe `javax.microedition.io.Connector` conter métodos de segurança, a classe `javax.bluetooth.RemoteDevice` também contém métodos relacionados com segurança. A classe *RemoteDevice* contém métodos que podem ser solicitados a qualquer momento para alterar a segurança da ligação ou para obter as definições de segurança existentes na ligação.

A Tabela 16 mostra os métodos de segurança pertencentes à classe *RemoteDevice* [*NOKIA developer*].

**Tabela 16.** Métodos da Classe *RemoteDevice*

| Classe | Métodos | | |
|---|---|---|---|
| RemoteDevice | authenticate | authorize | Encrypt |
| | getBluetoothAddress | getFriendlyName | getRemoteDevice |
| | hashCode | isAuthenticated | isAuthorized |
| | isEncrypted | isTrustedDevice | |

#### 4.4.2.3. Segurança em Aplicações de Servidor

Esta secção mostra um extrato de código fonte que tem como objetivo a ligação de cliente Bluetooth ao servidor Bluetooth, e os métodos de segurança com os parâmetros autenticação e encriptação ativos. Existem comentários dentro do próprio código de forma a compreender a sua execução.

Nas aplicações de servidor para portas série, o código contém parâmetros adicionais na *string* de ligação para indicar que a implementação deve efetuar autenticação e encriptação sempre que um cliente tente ligar-se a este serviço.

O seguinte código contém alguns comentários e mostra o procedimento de um servidor Bluetooth em espera por uma ligação de um cliente (código extraído de [*JSR-82*, 2002]):

```
String serversConnString =
     "btspp://localhost:3B9FA89520078C303355AAA694238F07;
      authenticate=true;encrypt=true";
      try {
           StreamConnectionNotifier notifier =
           (StreamConnectionNotifier)Connector.open(serversConnString);
           /* Esperar pela ligação do cliente. Nota: Se o cliente não conseguir autenticar-se ou se o link
para o cliente não consegue ser encriptado, a ligação é recusada pela API sem que a aplicação do servidor se
aperceba. */
          StreamConnection rfconn = (StreamConnection)notifier.acceptAndOpen();
      } catch (IOException e) {
}
```

#### 4.4.2.4. Segurança em Aplicações de Cliente

Nas aplicações de cliente para portas série, quando se liga ao servidor com o `Connector.open()`, o cliente utiliza parâmetros adicionais na *string* de ligação para estabelecer autenticação e encriptação.

O seguinte código contém comentários para explicar o procedimento de uma ligação e respetiva comunicação com dados encriptados (código extraído de [*JSR-82*, 2002]):

```
String encryptedMsg = "This message will be sent encrypted";
OutputStream os = null;
StreamConnection con = null;
ServiceRecord record;
/*
• Utiliza os métodos SDP do cliente para obter o ServiceRecord do servidor SDP.
• Define uma string solicitando que esta ligação ao serviço, descrita por serviço, seja autenticada e encriptada. O argumento de false significa que o cliente não necessita do role de master.
*/
String clientsConnString =
record.getConnectionURL(ServiceRecord.AUTHENTICATE_ENCRYPT,
false);
try {
     con = (StreamConnection)
     Connector.open(clientsConnString);
/* Caso se chegue a este ponto, significa que o dispositivo do servidor foi autenticado, e que todas as comunicações entre o cliente e o servidor através de con estão a ser encriptadas. */
     os = con.openOutputStream();
     /* O próximo passo envia os dados encriptados para o servidor. */
     os.write(encryptedMsg.getBytes());
     os.close();
} catch (BluetoothConnectionException e1) {
/* Caso o servidor não consiga autenticar-se ou se a ligação não consiga ser encriptada, será então lançado esta excepção. */
} catch (IOException e) {
    System.out.println(e.getMessage());
} finally {
    if (con != null) {
      try {
            con.close();
      } catch (Exception e) {
      }
   } }
```

#### 4.4.2.5. Serial Port Profile

Um servidor de *Serial Port Profile* (SPP) é uma aplicação que fornece serviços baseados no SPP. Uma aplicação que solicita uma ligação ao SPP é um cliente SPP. Tanto as aplicações de servidor ou de cliente do SPP podem estar em qualquer uma das extremidades da ligação de RFCOMM. O servidor SPP regista os serviços no *Service Discovery Database* (SDDB), sendo que o processo de registo é efetuado pela implementação (de *software*) que adiciona um *server channel identifier* ao registo do serviço.

Um cliente SPP localiza o serviço utilizando a API que tem como função a descoberta dos serviços. Após localizado, o cliente liga-se ao servidor especificando o endereço do servidor bem como o canal de identificação. A negociação dos parâmetros de ligação bem como o controlo de fluxo de dados entre os dispositivos são controlados automaticamente pelo SPP.

Um servidor SPP terá de inicializar os serviços que dispõe e registá-los no SDDB. Os seguintes objetos representam um serviço da porta série:

- `javax.microedition.io.StreamConnectionNotifier`: Este objeto escuta ligações de clientes para este serviço.
- `javax.bluetooth.ServiceRecord`: Este objeto descreve o serviço e como o serviço pode ser acedido.

Uma aplicação de servidor usa o método `Connector.open()`com uma ligação *Uniform Resource Locator* (URL) de servidor SPP para criar ambos os objetos mencionados acima, representando assim o serviço da porta série.

Ao invocar o `Connector.open()` com uma ligação URL de servidor SPP, este retorna o `StreamConnectionNotifier` que representa o serviço SPP. A Figura 29 mostra estas ligações e a comunicação com outros dispositivos de Bluetooth.

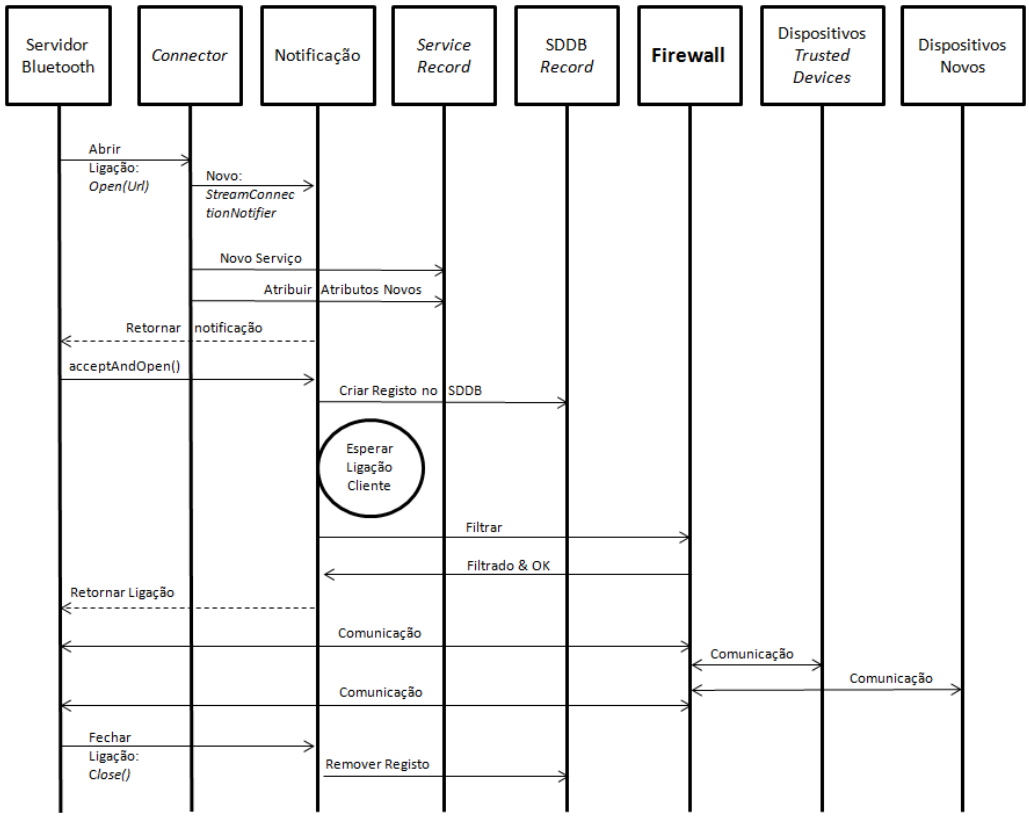



**Figura 29.** Ligação do servidor Bluetooth com registo do serviço e filtro da Firewall.

#### 4.4.2.6. Estabelecer Ligação através do Servidor

Tendo em conta a funcionalidade do SPP explicado na secção anterior, o servidor SPP cria o objeto `StreamConnectionNotifier` utilizando a *string* apropriada como argumento para o `Connector.open()` e fazendo o *cast* do resultado de `Connector.open()` para o `StreamConnectionNotifier`.

O seguinte código ilustra um exemplo de como estabelecer uma ligação através do servidor (código extraído de [*JSR-82*, 2002]):

```
StreamConnectionNotifier service =
       (StreamConnectionNotifier)Connector.open(
       “btspp://localhost:102030405060708090A1B1C1D1D1E100;name=SPPEx”);
StreamConnection con =
       (StreamConnection) service.acceptAndOpen();
```

Através do código acima referido, observa-se que o servidor utiliza o método `acceptAndOpen()` para aceitar uma ligação por parte do cliente. Este método fica em espera até que um cliente se ligue.

Quando o serviço aceita a ligação por parte do cliente, o `acceptAndOpen()` retorna o objeto `StreamConnection`.

Para que a implementação do `acceptAndOpen()` notifique o `btspp`, este tem de obrigar o *stack* do Bluetooth a enviar todas as comunicações entre a aplicação do cliente e a aplicação do servidor através de *streams* associadas com o objeto retornado pelo `acceptAndOpen()`. Este objeto retornado pelo `acceptAndOpen()` tem de implementar a interface `StreamConnection`.

O serviço SPP pode aceitar múltiplas ligações de diferentes clientes através do `acceptAndOpen()`, sendo que um `StreamConnection` é criado por cada ligação aceite. Estas ligações são efetuadas através do canal RFCOMM.

Caso o dispositivo de Bluetooth em questão não suporte múltiplas ligações, a implementação do `acceptAndOpen()` lança a exceção `BluetoothStateException`. O método `close()` é utilizado para fechar a ligação.

O método `close()` tem as seguintes caraterísticas:

- Quando utilizado antes da ligação ao serviço, o serviço associado ao `StreamConnectionNotifier` fica inacessível a clientes através da funcionalidade de descoberta de serviços.
- A implementação tem de remover o registo do serviço da SDDB ou inibir funcionalidades que o *stack* do Bluetooth fornece para que o serviço fique registado no SDDB mas inacessível a clientes.
- Será conveniente fechar o canal RFCOMM só quando todos os `StreamConnection` ao serviço bem como o próprio `StreamConnection` tenham sido fechados.

#### 4.4.2.7. Estabelecer Ligação através do Cliente

De forma a que um cliente SPP se ligue a um serviço SPP, este terá de procurar o serviço através da funcionalidade de descoberta de serviços. A URL de ligação do cliente contém o endereço do dispositivo de Bluetooth servidor e o identificador do canal do servidor para o serviço pretendido. Para obter a URL de

ligação de cliente para o serviço, é utilizado o método `getConnectionURL()` dentro do `ServiceRecord`.

O seguinte código mostra um exemplo de como um cliente estabelece ligação a um serviço com o canal de servidor com valor `5`, num dispositivo com endereço `0050C000321B`. Neste caso, o método `Connector.open()` é invocado retornando o objeto `StreamConnection` que representa a ligação SPP por parte do cliente (código extraído de [*JSR-82*, 2002]).

```
StreamConnection con =
     (StreamConnection)
          Connector.open(“btspp://0050C000321B:5”);
```

# 5. CONCLUSÃO E TRABALHO FUTURO

O Bluetooth é uma tecnologia usada em muitos dispositivos com ligações sem fio de curto alcance. Cresceu a um ritmo incrível desde o seu início e encontra-se implementado em milhões de objetos do dia-a-dia como é o caso das impressoras, auscultadores, leitores MP3, telemóveis, entre outros.

Dada a sua grande prevalência nestes dispositivos de massa, as questões relacionadas com a sua segurança assumem um relevo deveras importante. As três grandes áreas em que pensamos que a segurança é mais importante para esta tecnologia é na indústria automóvel, aplicações militares, e acima de tudo, telemóveis onde milhões de pessoas utilizam o Bluetooth para todo o tipo de tarefas.

Neste trabalho foram analisados os problemas de Bluetooth em termos de segurança e analisaram-se os ataques mais perigosos para os utilizadores através dos seus métodos e procedimentos, tendo o *BlueSnarf*, *BlueSnarf++*, e o *BlueBug* como os mais relevantes.

Os ataques *BlueSnarf* e *BlueBug* são capazes de obter informações pessoais como o calendário, a agenda e SMS. O *BlueSnarf++*, por outro lado é um aperfeiçoamento do *BlueSnarf* com a particularidade de permitir leitura e escrita no sistema de ficheiros do dispositivo. O *BlueBug* além de permitir acesso a dados pessoais do utilizador, permite ainda que um atacante possa fazer chamadas, reencaminhar chamadas, enviar SMS, conectar-se a redes sem fio, e até mesmo mudar o fornecedor de serviço do telefone.

Os ataques classificados nesta pesquisa foram observados entre os anos de 2003 a 2007. Em 2007, a versão Bluetooth 2.1 implementou o Modo de Segurança 4 que é utilizado no SSP, mas este modo de segurança não protege contra o ataque *man-in-the-middle*. Embora o Bluetooth tenha implementado modos de segurança para tentar proteger o utilizador contra ataques, existem novos tipos de ataques como vírus e *malwares* que são os novos desafios de proteção, tais como *Cabir*, o *Flame* e o *Obad.a*.

Todos os ataques que foram estudados e que afetam a segurança do Bluetooth podem ser mitigados ao proteger o dispositivo a partir do protocolo

RFCOMM, já que é através deste protocolo que a comunicação é efetuada. Para proteger este protocolo é proposto uma *firewall* para Bluetooth. Esta *firewall* filtra as ligações de RFCOMM e associa-as a um perfil definido pelo seu utilizador. Este método é baseado numa *firewall* WAF que protege contra ligações ou conteúdos nocivos ao nível das aplicações Web, protegendo assim o utilizador de qualquer tentativa de ataque antes que este possa interagir com a aplicação. Tal como a WAF, todas as ligações serão filtradas pela *firewall* do Bluetooth, mesmo que o dispositivo Bluetooth que esteja a ligar-se faça parte da lista de *trusted devices*, e mesmo fazendo parte de um perfil, a *firewall* filtra o endereço MAC e o UUID de cada dispositivo para verificar se de facto o dispositivo que está a ligar-se é seguro, ou se é necessário atribuir um perfil para a sua utilização. Ao aplicar este tipo de filtro no ponto de entrada das ligações, os ataques ao Bluetooth podem ser impedidos.

A associação de cada dispositivo Bluetooth a um perfil permite que o grau de segurança aumente visto que cada ligação efetuada terá as restrições de utilização pré-definidas. Os três perfis definidos por omissão cobrem a maioria dos casos para a utilização habitual do Bluetooth: *@Home* - comunicação de *inbound* e *outbound* sem limitação; *Temporary* - comunicação só de *inbound* e dispositivo Bluetooth removido automaticamente após um determinado período de tempo; *E-Commerce* – com maiores cuidados de segurança e restrições de acesso à informação existente no dispositivo móvel. Ao separar as ligações de Bluetooth por perfis, estas ligações não podem interagir com outros perfis devido às regras que são impostas especificamente para cada perfil. No caso do *E-Commerce*, este perfil é muito restrito em termos de regras, sendo estas regras essenciais para a identificação do dispositivo de forma a interagir depois com a plataforma de pagamento que seguirá as regras de IPC DSS.

A plataforma proposta para o desenvolvimento da *firewall* é o J2ME que é baseada na linguagem Java, já que permite funcionar na maioria dos telemóveis, nomeadamente os que são baseados em Android. Para esta plataforma de desenvolvimento, descreveram-se as classes e métodos relativos às ligações e segurança, focando primordialmente nas partes essenciais para lidar com as ligações Bluetooth.

A *firewall* terá a particularidade de o utilizador poder criar novos perfis para além dos perfis que vêm por omissão, fornecendo assim uma maior diversidade de utilização de acordo com as necessidades do utilizador.

Será também possível escolher se o utilizador pretende ativar ou não o *Stealth Mode*, mecanismo este que ajuda a impedir que utilizadores mal-intencionados descubram informações sobre o dispositivo móvel e serviços que são executados. Embora o Bluetooth já tenha uma opção de visibilidade do dispositivo que é parecido ao *Stealth Mode*, convém ter um mecanismo que seja controlado pela *firewall*.

A continuação natural do trabalho que foi desenvolvido é um protótipo que servirá simultaneamente para validar a proposta apresentada e para avaliar o seu desempenho. Para além dos aspectos já discutidos no relatório, há outros que prevemos que sejam necessário aprofundar. Os mais importantes dos quais se encontram a seguir.

A forma exacta como irá ser implementada a filtragem dos dados de entrada (*inbound*) e de saída (*outbound*) da *firewall* é um aspecto muito importante. Há questões de desempenho e do sistema operativo que terão de ser analisados tendo em conta o método de implementação. O modo como a *firewall* filtra os dados é muito importante e terá de ser decidido.

Um dos métodos de filtragem dos dados pode ser do tipo DPI, em que a *firewall* filtra os pacotes pelo tipo de dados desde que estes não sejam encriptados.

Outra possibilidade de filtragem por parte da *firewall* será utilizando um método tipo WAF, em que a *firewall* filtra o tipo de aplicações que poderão ser acedidas. Esta filtragem poderá ser feita através da análise do conteúdo dos pacotes e não apenas do seu cabeçalho, desde que estes dados não sejam encriptados.

Uma outra estratégia será a da *firewall* filtrar as aplicações que serão acedidas, e impedir que determinada aplicação possa aceder a recursos não autorizados.

Para se conseguir que a *firewall* cumpra todas as especificações, ela terá de ter um acesso privilegiado (direto) a funções internas do sistema operativo. Isto poderá ser feito através da instrumentalização de APIs do próprio sistema operativo de modo a que a *firewall* possa ter acesso a informações de controlo, e possa também impedir acesso a recursos internos do dispositivo.

Em termos de desenvolvimentos futuros da *firewall*, podemos propor o acrescento do protocolo de *audio* já que este não utiliza o RFCOMM para comunicação,

logo não passará pela *firewall* (tal como está definida). A relevância desta inclusão na *firewall* prende-se com o áudio ser das funcionalidades mais utilizadas pelos utilizadores, por exemplo, kit mãos livres para automóveis, auriculares para telemóveis e auscultadores.

# REFERÊNCIAS BIBLIOGRÁFICAS

## Livros

Abdel-Aziz, A. (2009), "Intrusion Detection & Response, Leveraging Next Generation Firewall Technology", SANS Institute. Acedido a 1 de junho 2014, em: http://www.sans.org/reading_room/whitepapers/firewalls/rss/intrusion_detection_and_response_leveraging_next_generation_firewall_technology_33053.

Bluetooth Specification Version 4.0, "BLUETOOTH SPECIFICATION Version 4.0" [Vol 0]. Master Table of Contents & Compliance Requirements. Publication date: 17 December 2009.

C Bala Kumar et al. Motorola (2003), "C Bala Kumar, Paul J. Kline & Timothy J. Thompson. Bluetooth Application Programming with the Java APIs". Published By Morgan Kaufmann Publishers, (c) Motorola, Inc. 2003.

Erik Dahlman et al. (2008), "Erik Dahlman, Stefan Parkvall, Johan Shold, Per Beming. 3G Evolution, HSPA and LTE for Mobile Broadband". Second Edition, 2008.ISBN: 978-0-12-374538-5, Elsevier.

JSR-82 (2002), "JavaTM APIs for Bluetooth TM Wireless Technology (JSR-82)". Specification Version 1.0a, April 5, 2002

Karen Scarfone et al. (2008), "Karen Scarfone, John Padgette. Guide to Bluetooth Security". National Institute of Standards and Technology. Publication 800-121. September 2008.

Karen Scarfone et al. (2012), "Karen Scarfone, John Padgette, Lily Chen. Guide to Bluetooth Security". National Institute of Standards and Technology. Publication 800-121 Revision 1. June 2012.

Naval Education And Training (1998), "Navy Electricity and Electronics Training Series". Module 17-Radio-Frequency Communications Principles NONRESIDENT TRAINING COURSE SEPTEMBER 1998

NOKIA AT Commands (2000), "Nokia PremiCell Data List of AT Commands". 9351671, Issue 4. NOKIA, 2000.

PCI DSS (2010), Payment Card Industry (PCI) Data Security Standard. Requirements and Security Assessment Procedures, Version 2.0. October 2010. Acedido a 6 de julho 2014, em: https://www.pcisecuritystandards.org/documents/pci_dss_v2.pdf.

## Comunicações em atas de conferência

Adam Laurie et al. (2004), "Adam Laurie, Marcel Holtmann, Martin Herfurt. Hacking Bluetooth enabled mobile phones and beyond – Full Disclosure. 21C3: The Usual Suspects". 21st Chaos Communication Congress December 27th to 29th, 2004.

Adam Laurie et al. (2005), "Adam Laurie, Marcel Holtmann and Martin Herfurt. Bluetooth Hacking The State of the Art". 22C3 December 30st 2005, Berlin, Germany.

Adam Laurie, 2006, "Adam Laurie. Digital detective – Bluetooth". 2006 Elsevier.

Andrew Y. Lindell (2008), "Attacks on the Pairing Protocol of Bluetooth v2.1". Aladdin Knowledge Systems and Bar-Ilan University, Israel. June 25, 2008.

João Alfaiate et al. (2012), "João Alfaiate and José Fonseca, Bluetooth Security Analysis for Mobile Phones", Information Systems and Technologies (CISTI), 2012, 7th Iberian Conference.

José Fonseca et al. (2009), "Fonseca, J., M. Vieira, and H. Madeira (2009), Vulnerability & Attack Injection for Web Applications", in IEEE International Conference on Dependable Systems and Networks with FTCS and DCC, 2009. DSN 2009.

Kevin Streff et al. (2009), "Kevin Streff, Justin Haar. An Examination of Information Security in Mobile Banking Architectures". Dakota State University. Journal of Information Systems Applied Research. June 10, 2009.

Mohamed GHALLALI et al. (2011), "Mohamed GHALLALI, Driss EL OUADGHIRI, Mohammad ESSAAIDI, Mohamed BOULMALF. Mobile Phones Security: The Spread of Malware via MMS and Bluetooth, Prevention Methods". MoMM2011, 5-7 December, 2011, Ho Chi Minh City, Vietnam.

Terrence Oconnor et al. (2008), "Terrence OConnor, Douglas Reeves. Bluetooth Network-Based Misuse Detection". NC State University Raleigh. 2008 Annual Computer Security Applications Conference, 2008 IEEE.

Yaniv Shaked et al. (2005), Yaniv Shaked and Avishai Wool. "Cracking the Bluetooth PIN". MobiSys '05: The Third International Conference on Mobile Systems, Applications, and Services. Acedido a 1 de junho 2014, em: https://www.usenix.org/legacy/event/mobisys05/tech/full_papers/shaked/shake

d.pdf /.

Yu Xin et al. (2009), “Yu Xin, Yan Ting. A Security Architecture Based on User Authentication of Bluetooth”. College of Automation, Beijing Union University, Beijing 100101, China. © 2009 IEEE.

## Artigos em revista

DELL (2012), "How Traditional Firewalls Fail Today’s Networks — And Why Next-Generation Firewalls Will Prevail". © 2012 Dell SonicWALL. ”. Acedido a 25 de junho 2014, em: http://software.dell.com/documents/how-traditional-firewalls-fail-todays-networks-ebook-24532.pdf.

CSI (2009), “14th Annual CSI Computer Crime and Security Survey”. Executive Summary, December 2009.

Esteban Alcorn (2011), “Bluetooth Architecture Overview”. Windows Embedded Compact 7. March 2011 © Microsoft.

R. W. SIMONS (1996), “Guglielmo Marconi and Early Systems of Wireless Communication”. GEC REVIEW, VOL. 11, NO. 1, 1996

Symantec (2008), “Symantec Report on the Underground Economy”. Published November 2008.

## Tese de Doutoramento

José Fonseca (2011), "Evaluating the [In]security of Web Applications". Thesis for the degree of Doctor of Philosophy. Department of Informatics Engineering Faculty of Sciences and Technology University of Coimbra.

## Tese de Curso

Andreas Becker (2007), “Andreas Becker, Bluetooth Security & Hacks”. Seminararbeit, Ruhr-Universitat Bochum. August 16, 2007. Acedido a 24 de maio 2014 http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.392.8834&rep=rep1&type=pdf.

## Documento eletrónico disponível na Internet

3gpp, Mobile Broadband Standard, “The Mobile Broadband Standard”. Acedido a 1 de junho 2014, em: http://www.3gpp.org/technologies/keywords-acronyms/102-gprs-edge.

Alastair Stevenson (2013), “Android mobile malware using Bluetooth to sneak onto

Google smartphones". June 26, 2013. Acedido a 1 de junho 2014, em: http://www.v3.co.uk/v3-uk/news/2277556/android-mobile-malware-using-bluetooth-to-sneak-onto-google-smartphones.

Anindya Bakshi (2007), "Anindya Bakshi, MindTree Consulting. Bluetooth Secure Simple Pairing". December 2007. Acedido a 20 de maio 2014, em: http://www.wirelessdesignmag.com/PDFs/2007/1207/wd712_coverstory.pdf.

Apple Feature - Stadiums (2013), "Apple Feature to Turn MLB Stadiums Into Interactive Playgrounds". 27.Septemer.2013. Acedido a 1 de junho 2014, em: http://mashable.com/2013/09/26/mlb-at-the-ballpark-app/.

AT Commands, "AT Commands, GSM AT command set". Acedido a 29 de junho 2014, em: http://www.engineersgarage.com/tutorials/at-commands.

BackTrack-linux. Acedido a 1 de junho 2014, em: http://www.backtrack-linux.org.

Bank of America (2010), "Bank of America. Mobile Banking Available to More Than 20 Million Online Customers Across the U.S". Acedido a 20 de maio 2014, em: http://newsroom.bankofamerica.com/index.php?s=43&item=7772.

banktech (2013), "Bank of America Tops 1 Million Small Business Mobile Banking Users". August 20, 2013. Acedido a 1 de junho 2014, em: http://www.banktech.com/channels/bank-of-america-tops-1-million-small-bus/240160237.

BBC News TECHNOLOGY (2012), "UN: Six billion mobile phone subscriptions in the world". Acedido a 1 de junho 2014, em: http://www.bbc.com/news/technology-19925506.

BBC News TECHNOLOGY (2013), "Mobile phone celebrates 40th anniversary". 3 April 2013. Acedido a 1 de junho 2014, em: http://www.bbc.com/news/technology-22013228.

Bluetooth beacons (2014), "Bluetooth beacons signal future public safety apps and advantages". Jan 15, 2014. Acedido a 1 de junho 2014, em: http://gcn.com/articles/2014/01/15/bluetooth-beacons.aspx.

Bluetooth Product Directory, "Bluetooth SIG". Acedido a 1 de junho 2014, em: http://www.bluetooth.com/Pages/Product-Directory.aspx.

Bluetooth SIG, "Basics, Bluetooth Special Interest Group". Acedido a 14 de maio 2014, em: http://www.bluetooth.com/English/Technology/Pages/Basics.aspx.

Bluetooth SIG, “History of Bluetooth Technology, Bluetooth Special Interest Group”. Acedido a 1 de junho 2014, em: http://www.bluetooth.com/Pages/History-of-Bluetooth.aspx.

BlueZ. Acedido a 1 de junho 2014, em: http://www.bluez.org/about/.

boston.com (2009), “To test high-speed 4G cellular network”, August 13, 2009. Acedido a 1 de junho 2014, em: http://www.boston.com/business/technology/articles/2009/08/13/in_test_of_4g_network_hub_to_get_early_look_at_next_level_web_link/.

Calcular uma CAGR. Acedido a 1 de junho 2014, em: http://office.microsoft.com/pt-br/excel-help/calcular-uma-cagr-taxa-composta-de-crescimento-anual-HP010070476.aspx.

Calvin Azuri (2010), “Calvin Azuri. TMCnet. Mobile E-Commerce Expected to Increase 65 Percent Annually through 2015”. April 28, 2010. Acedido a 1 de junho 2014, em: http://www.tmcnet.com/news/2010/04/28/4756408.htm.

Catherine Roseberry, About.com, “Don't drive and use your cell phone here”. Acedido a 1 de junho 2014, em: http://mobileoffice.about.com/cs/traveladvice/qt/usingcellphone.htm.

Charles Hodgdon (2003), “Ericsson Technology Licensing. Adaptive Frequency Hopping for Reduced Interference between Bluetooth® and Wireless LAN”. Acedido a 1 de junho 2014, em: http://www.design-reuse.com/articles/5715/adaptive-frequency-hopping-for-reduced-interference-between-bluetooth-and-wireless-lan.html.

Cnet (2013), “LTE users to hit 1 billion by 2016”. January 22, 2013. Acedido a 1 de junho 2014, em: http://news.cnet.com/8301-1035_3-57565310-94/lte-users-to-hit-1-billion-by-2016-says-report/.

Dawn (2009), “Billion plus’ mobile users in India by 2015”. Nov 18, 2009. Acedido a 1 de junho 2014, em: http://www.dawn.com/news/935612/billion-plus-mobile-users-in-india-by-2015-telecom.

Defense in Depth, NSA. “A practical strategy for achieving Information Assurance in today’s highly networked environments”. Acedido a 1 de junho 2014, em: http://www.nsa.gov/ia/_files/support/defenseindepth.pdf.

die.net, sdptool, “Linux man page”. Acedido a 1 de junho 2014, em: http://linux.die.net/man/1/sdptool.

Don Reisinger (2007), “CNET, Bluejacking, bluesnarfing and other mobile woes”.

August 23, 2007. Acedido a 1 de junho 2014, em: http://www.cnet.com/news/bluejacking-bluesnarfing-and-other-mobile-woes/.

Eric Geier (2011), "Intro to Next Generation Firewalls". Acedido a 27 de junho 2014, em: http://www.esecurityplanet.com/security-buying-guides/intro-to-next-generation-firewalls.html.

F-Secure, "Bluetooth-Worm: SymbOS/Cabir". Acedido a 1 de junho 2014, em: http://www.f-secure.com/v-descs/cabir.shtml.

GIGAOM (2010), "Verizon's 4G LTE Service Arrives Dec. 5 With 3G Prices". DEC. 1, 2010. Acedido a 1 de junho 2014, em: http://gigaom.com/2010/12/01/verizon-lte-4g-launch/.

Global Thoughtz Mobile (2009), "Global Thoughtz Mobile. India will touch 1 Billion mobile subscribers very soon". November 18, 2009. Acedido a 14 de Maio 2014, em: http://mobile.globalthoughtz.com/index.php/india-will-touch-1-billion-mobile-subscribers-very-soon.

GSM Arena. Acedido a 1 de junho 2014, em: http://www.gsmarena.com/apple_iphone_5s-5685.php.

hcitool (2002), "Linux Command". Nov 12, 2002. Acedido a 1 de junho 2014, em: http://linuxcommand.org/man_pages/hcitool1.html.

iClarified (2011), "Skype iOS App is Vulnerable to Attack That Can Steal Your Address Book". Acedido a 1 de junho 2014, em: http://www.iclarified.com/entry/index.php?enid=17003.

iClarified (2014), "Apple SVP Phil Schiller Points to Cisco Report That Found 99% of Mobile Malware Targets Android". January 21, 2014. Acedido a 1 de junho 2014, em: http://www.iclarified.com/37695/apple-svp-phil-schiller-points-to-cisco-report-that-found-99-of-mobile-malware-targets-android.

IC INSIGHTS (2012), "Unit Shipments of Bluetooth-Enabled Equipment to Double by 2015". JUNE 21, 2012. Acedido a 1 de junho 2014, em: http://www.icinsights.com/news/bulletins/Unit-Shipments-Of-BluetoothEnabled-Equipment-To-Double-By-2015/.

IDC - Press Release (2013), "Smartphone Shipments Third Quarter of 2013". Acedido a 30 de março 2014, em: http://www.idc.com/getdoc.jsp?containerId=prUS24418013.

IDC - Press Release (2010), "Worldwide Converged Mobile Device". Acedido a 22 de maio 2010, em: http://www.idc.com/getdoc.jsp?containerId=prUS22333410.

IDC - Smartphone OS (2014), "Android and iOS Continue to Dominate the Worldwide Smartphone Market". February 12, 2014. Acedido a 29 de junho 2014, em: http://www.idc.com/getdoc.jsp?containerId=prUS24676414.

iOS Bluetooth profiles (2013), “iOS Supported Bluetooth profiles”. Oct 8, 2013. Acedido a 1 de junho 2014, em: http://support.apple.com/kb/ht3647.

Jim McMillan (2009), "Intrusion Detection FAQ: What is the difference between an IPS and a Web Application Firewall?". Acedido a 28 de junho 2014, em: http://www.sans.org/security-resources/idfaq/ips-web-app-firewall.php.

John P. Mello Jr. (2013), “Android Trojans spread by Bluetooth, hijack bank codes”. June 11, 2013. Acedido a 1 de junho 2014, em: http://www.networkworld.com/news/2013/061013-android-trojans-spread-by-bluetooth-270695.html.

Keith Regan (2006), “CBS to Use Bluetooth to Beam TV Clips to Passersby”. August 25, 2006. Acedido a 1 de junho 2014, em: http://www.ecommercetimes.com/story/52649.html?wlc=1273673468.

Li Weitao - CHINA daily (2006), “Telecom industry on a roll”. 2006-12-04. Acedido a 1 de junho 2014: http://www.chinadaily.com.cn/bizchina/2006-12/04/content_749312_2.htm.

Lynn Tan (2007), “ZDNet, Symantec warns users over Bluetooth security”. September 21, 2007. Acedido a 1 de junho 2014, em: http://www.zdnet.com/news/symantec-warnsusers-over-bluetooth-security/165841.

Mathew J. Schwartz (2012), “Flame Taps Bluetooth: Security Implications”. Acedido a 1 de junho 2014, em: http://www.informationweek.com/mobile/flame-taps-bluetooth-security-implications/d/d-id/1104640.

Media Ford, (2007), “Media Ford. Ford Teams up with Microsoft to deliver SYNC; In-car Digital System Exclusive to Ford”. Jan 7, 2007. Acedido em 23 de fevereiro 2012, em: http://media.ford.com/Article_Display.Cfm?Article_Id=25168.

Neil Roiter, Bluetooth 2.1 is easy to crack. 07 Aug, 2008. Acedido a 1 de junho 2014, em: http://searchsecurity.techtarget.com/news/article/0,289142,sid14_gci1324335,00.html.

NOKIA developer. Acedido a 1 de junho 2014, em: http://developer.nokia.com/Resources/Library/Java/_zip/GUID-9F75713D-5642-4C39-9A33-C20928F37BF7/javax/bluetooth/RemoteDevice.html.

nruns professionals, “Security Tools - BTcrack 1.1”. Acedido a 14 de maio 2010, em: http://www.nruns.com/_en/security_tools_btcrack.php.

Palowireless, L2CAP, “Bluetooth Resource Center. Logical Link And Adaptation Layer”. Acedido a 8 de novembro 2013, em: http://www.palowireless.com/Bluearticles/adapt.asp.

Palowireless, SDP, “Palowireless, Bluetooth Resource Center. Service Discovery Protocol”. Acedido a 10 de novembro 2013, em: http://www.palowireless.com/infotooth/tutorial/sdp.asp.

Parrot, wireless devices, “Wireless devices for mobile phones”. Acedido em 23 de fevereiro 2012, em: http://www.parrot.com/usa/.

Pentest, Security Consultancy. Acedido a 18 de maio 2010, em: http://www.pentest.co.uk/cgibin/viewcat.cgi?cat=downloads&section=01_bluetooth.

Ricky Panchal (2005), "Firewalls: Hardware vs. Software". Acedido a 27 de junho 2014, em: http://www4.ncsu.edu/~kksivara/sfwr4c03/projects/4c03projects/RPanchal-Project.pdf.

Roman Unuchek (2013), “Trojan now being distributed via mobile botnets”. September 05, 2013. Acedido a 1 de junho 2014, em: https://www.securelist.com/en/blog/8131/Obad_a_Trojan_now_being_distributed_via_mobile_botnets.

Samsung Galaxy. Acedido a 1 de junho 2014, em: http://www.samsung.com/pt/consumer/mobile-phone/note/note/SM-N9005ZWETPH.

Security SW, “Security and Privacy Software”. Acedido a 27 de junho 2014, em: http://www.security-and-privacy-software.com/ip-blacklist-versus-ip-whitelist.html.

SIG - Adopted Specifications. © 2014 Bluetooth SIG, Inc. Acedido a 1 de junho 2014, em: https://www.bluetooth.org/en-us/specification/adopted-specifications.

Software.informer, Bluescanner. Acedido a 1 de junho 2014, em: http://bluescanner.software.informer.com/.

Trifinite Group. Acedido a 1 de junho 2014, em: http://trifinite.org/trifinite_group.html.

Trifinite Stuff. Acedido a 1 de junho 2014, em: http://trifinite.org/trifinite_stuff.html.

UN News Center (2008), “UN News Center. Number of cell phone subscribers to hit 4 billion”. Sep 25, 2008. Acedido em 30 de março 2014, em: http://www.un.org/apps/news/story.asp?NewsID=28251.

Watchguard, "Next-Generation Firwall". Acedido a 29 de junho 2014, em: http://www.watchguard.com/wgrd-products/ngfw/overview.

Webdesignerdepot (2009), “The Evolution of Cell Phone Design Between 1983-2009”. May 22, 2009. Acedido a 1 de junho 2014, em: http://www.webdesignerdepot.com/2009/05/the-evolution-of-cell-phone-design-between-1983-2009/.

William Jackson (2012), “NIST issues guide to fixing the holes in Bluetooth”. Jun 13, 2012. Acedido a 1 de junho 2014, em: http://gcn.com/articles/2012/06/13/nist-bluetooth-security-guide.aspx.

Windows Profiles (2011), "Understanding Firewall Profiles". Windows Server. Acedido a 28 de junho 2014, em: http://technet.microsoft.com/en-us/library/getting-started-wfas-firewall-profiles-ipsec(v=ws.10).aspx.

Windows Rules (2009), "Understanding Firewall Rules". Windows Server. Acedido a 28 de junho 2014, em: http://technet.microsoft.com/en-us/library/dd421709(v=ws.10).aspx.

Wireless and Mobile News (2009), “Wireless and Mobile News. Mobile Apps Race to Serve Auto Market @IAA”. Oct 12, 2009. Acedido em 23 de fevereiro 2012, em: http://www.wirelessandmobilenews.com/2009/10/mobile-apps-race-to-serve-auto-market-iaa-says-isuppli.html.

# ANEXO A – ARTIGO CISTI 2012

Este anexo apresenta o artigo científico *Bluetooth Security Analysis for Mobile Phones* e que foi publicado no CISTI'2012 (*7th Iberian Conference on Information Systems and Technologies*) em Madrid, Espanha, e apresentado no dia 20 junho de 2012.

# Bluetooth Security Analysis for Mobile Phones

João Alfaiate, José Fonseca
UDI – Research Unit for Inland Development of Guarda Polytechnic Institute, Portugal
jc.alfaiate@gmail.com, josefonseca@ipg.pt

***Abstract*— Mobile devices are becoming more and more omnipresent due to their lightweight, small size and increasing performance. Almost every mobile device has Bluetooth (BT) capabilities and this powerful combination widely used in our daily life is coming to new environments like the car and the military industries. As any technology, BT has security issues that hackers have extensively exploited over the years, while users seem not to care too much. To raise the security awareness we present an analysis of BT attack methods and tools over time. We paid particular attention to the severity, possible targets and the ability to persist over new versions of BT. Results show that adversaries can take complete control of the victims' mobile device features if they forget to use simple safety measures like turning off the BT when not in use. To increase security we also propose the development of a novel BT Firewall.**

***Keywords-Mobile Phones, Security, Hack, Attack, Bluetooth.***

## I. INTRODUCTION

Mobile phones are increasingly becoming omnipresent in our lives. They evolved from simple devices that could only be used to make phone calls and send short text messages to fully featured miniature computers. Nowadays, they are capable of browsing the Internet, read and send emails, edit documents, perform complex calculations, synchronize calendars and to-do lists, take photos, make videos, play games, and much more. This may justify the 6 billion mobile phone users, representing 87% of the world population [1].

The huge adoption of mobile phones followed the implementation of new developments in technology. One such technology that plays an important role is Bluetooth (BT) that is used to share contacts, create personal networks, hands-free communication, and much more.

The growing acceptance of mobile devices and their new features allowed them to invade markets that were not foreseen when they were first developed. Evidences of this can be found in the car industry, the military [2], all sorts of multimedia, advertising, and in daily tasks like e-Commerce and e-Banking [2]. Nevertheless, every time a new technology reaches the masses like this, it also calls the attention and the interest of malicious minds that want to exploit this new opportunity on their benefit.

In this paper, we present a study analyzing the BT presence in mobile devices, focusing on their security problems. We address this important aspect by analyzing common mobile attacks using BT and how to prevent them. We paid special attention to the most critical security problems affecting BT, which are BlueSnarf, BlueSnarf++ and BlueBug.

BT is a common entry point for many attack methods in mobile phones, and the information presented in this paper can be useful to educate and raise the awareness of mobile users in order to follow best practices. Using the right application configuration and simple things like turning off BT when not in use is essential to avoid the possibility of attacks. With over than 70% mobile phones with BT, they should offer a more reliable protection to their users [3]. In fact, security should not be pushed back to users and a mechanism should exist to prevent undesired access. This is why we also propose the development of a BT Firewall, which from the best of our knowledge does not exist yet.

The remaining of this paper is organized as follows: Section II presents the BT evolution over time. Section III details the BT security features. Section IV provides a study on BT attack methods and tools. Section V presents an example of a BT attack and the proposal of a BT Firewall. Section VI concludes the paper and introduces future work directions.

## II. BLUETOOTH EVOLUTION

Since the first public version of BT in 1999, five updated versions were released until 2010. They are shown in Table I, along with the most relevant features for our study [4, 5].

TABLE I. BLUETOOTH FEATURES

| Bluetooth Versions | Year | Faster connection | SSP | Security Mode 4 | Bug fixes | Error detection | Synchronization | Data Rate | L2CAP | HCI for AMP | Security for AMP | Power consumption |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1.1 | 2002 | | | | X | | | | | | | |
| 1.2 | 2003 | X | | | | X | X | | | | | |
| 2.0 | 2004 | | | | | | | X | | | | |
| 2.1 | 2007 | | X | X | | | | | | | | X |
| 3.0 | 2009 | | | | | | | X | X | X | X | X |
| 4.0 | 2010 | | | | | | | | | | | X |
| | | **New Features** | | | **Enhancement Features** | | | | | | | |

The most relevant BT enhancements were provided by the following versions:

- **Version 2.1**: Security aspects.
- **Version 3.0**: Enhanced Data Rate (EDR), which provides more speed and improved battery life.
- **Version 4.0**: Low power consumption.

PEst-OE/EGE/UI4056/2011 – project financed by Science and Technology Foundation

Data speed is a common concern and has been addressed in almost every version. The latest versions (2.1, 3.0 and 4.0) have a huge concern on power consumption, which is crucial for mobile devices. However, important security aspects were only effectively addressed with version 2.1 in 2007, eight years after BT was first released.

The following sub-sections provide details on data speed and the usage of BT in mobile phones, e-Business, car industry, and in the military. Because of its importance, security is addressed in its own section III.

*A. Data speed and radio ranges*

Since the release of BT, many manufactures used its reasonable data transfer rate ability for a wide range of purposes, such as printers, cameras, mobile phones (Personal Digital Assistance and Smartphones included), notebooks, among others.

Initially, BT started with 1 Mbps data rate and increased to 3 Mbps in version 2.0. However, the biggest leap occurred in version 3.0 allowing up to 24 Mbps, which was an 800% enhancement.

For the proper utilization of the device, these transfer rates are very important, but is also very important the distance range that can be reached. BT is divided in three radio ranges [5, 6]:

- **Class 1**: approximately 100 meters (300 feet).
- **Class 2**: approximately 10 meters (33 feet).
- **Class 3**: approximately 1 meter (3 feet).

*B. Bluetooth in mobile phones*

BT is used in many of our daily personal objects, but the most used scenario is probably for mobile communications. Mobile phones grew to be the best market for BT devices. With over than 70% of BT enabled devices being mobile phones [3] Nokia seems to lead the market share [7].

This major mobile boom allowed BT to increase its shipment in a yearly basis. Today, over than one billion BT devices are in use worldwide [8], and by the year 2013 the shipments expect to exceed 2 billion [9].

*C. Bluetooth in the car industry*

In some countries it is against the law to use a mobile phone while driving a car [10], but using a wireless system to communicate with a mobile phone is legal since the driver has both hands free. The car industry is quite interested in BT and many vehicles have BT capabilities. The following list details some features specifically developed for cars:

- Nokia Research Center presented a solution that allows the control of several electronic media systems of the car using BT [11]. The car displays the phone's applications and the driver can control them either by voice or by touch screen.
- Parrot SA presented an Android based head unit (the hardware component that interfaces electronic media systems with the car) that includes hands-free BT [11, 12].
- The Ford motor company in collaboration with Microsoft launched Ford SYNC, which is capable of connecting to any mobile phone or digital media player with the car itself [13].

*D. Bluetooth in the military*

The military has several projects that actually use BT as the communication protocol. Some examples are [2]:

- The Defense Advanced Research Project Agency (DARPA) with their wireless mesh network for the LANdroids project.
- The Air Force Research Laboratory (AFRL) for their group of miniature helicopters connected by BT.
- The Space and Naval Warfare Systems Center with their mobile robot which uses BT.

However, BT features and its ease of use can also jeopardize the security of the devices. For example, the US Navy tested a recruiting method using the BT data transfer ability. In 13 key locations with a population of 11,000 BT devices, the Navy could anonymously transfer videos to 18% of such devices [2]. If it was possible to send video it would also be possible to send any other file with malicious intentions.

When data transfer or third party resource access is necessary, security should be a top concern. This is discussed in the next section.

## III. BLUETOOTH SECURITY

The worldwide spread of mobile phones with BT and the decision to use it in situations not foreseen when the BT protocol was developed, attracted the attention for security problems. To address these security problems, in 2007 BT version 2.1 (the fifth release) had more security features than all the other versions, affecting a huge number of security related aspects [4]. Below are the most relevant:

- **Encryption Pause and Resume**: This feature pauses the encryption when the link key connection needs to be changed and when the master and slave roles of the devices need to be switched. After these changes, the encryption resumes.
- **Secure Simple Pairing (SSP)**: Created to simplify the pairing process and improve the BT security. The two main security aspects are to protect against passive eavesdropping and man-in-the-middle attacks. It uses the Elliptic Curve Diffie Hellman (ECDH) public key cryptography as a means to prevent passive eavesdropping attacks.
- **Security Mode 4**: Used for SSP.

BT is still one of the causes of security problems in mobile phones, in spite of the updates released. Even BT version 2.1 mostly devoted to security, still seems to leave some security problems unsolved. According to Andrew Lindell, chief cryptographer for Aladdin Knowledge Systems Ltd, SSP in version 2.1 has specification bugs allowing man-in-the-middle attacks [14]. For example, the six random digit password used in paring the devices can be obtained within 10 attempts. BT has four security modes to pair devices [6]:

- **Security Mode 1** allows unsecured links.
- **Security Mode 2** procedures are executed after the link establishment. This is a service level enforced

security where BT service security can be configured to use authentication and authorization, authentication only, or open to all devices.

- **Security Mode 3** initiates link-level security before the physical link is fully established.
- **Security Mode 4** is a service level security mode where the link-level connections are initiated only after link setup, but with added security due to the SSP. This mode is mandatory for all BT versions after 2.1 inclusive, but Security Mode 2 is used instead when the remote device does not support SSP.

The National Institute of Standards and Technology (NIST) consider Security Mode 3 as the strongest, due to authentication and encryption establishment requirement before the physical link is established [6]. To keep software that uses BT secure, organizations are advised to use the strongest security mode available for BT devices (Security Mode 3).

## IV. BLUETOOTH ATTACK ANALYSIS

This section presents BT security problems, and describes the most common procedures used to exploit BT vulnerabilities.

During our research, we found 6 tools and 11 methods to attack BT. Table II shows the attacks in a chronological point of view according to the year of discovery. We can verify that the majority of the attacks appeared between years 2004 and 2007. This timeframe corresponds to the upgrade of BT from version 2.0 to 2.1, which took three years to be released.

TABLE II. BT Attack PROCEDURES

| Year | Attack procedure | OS | | *Tool* | *Method* |
|---|---|---|---|---|---|
| | | *Linux* | *Windows* | | |
| 2001 | BTScanner | X | X | X | |
| 2003 | BlueSnarf | | | | X |
| | BlueJacking | | | | X |
| 2004 | Blooover | X | X | X | |
| | BlueBug | | | | X |
| | BlueSmack | | | | X |
| | Bluetooone | | | | X |
| | BlueSniper | | | | X |
| | Blueprinting | | | | X |
| 2005 | BlueSnarf++ | | | | X |
| | HeloMoto | | | | X |
| | Crack PIN | | | | X |
| | Car whispering | X | | X | |
| | HIDattack | | | | X |
| 2006 | BackTrack | X | | X | |
| | BlueScanner | | X | X | |
| 2007 | BTCrack | | X | X | |

From Table II we can verify that there are many ways to attack a BT device. We verify that the three most recent procedures are Tools, and two of them were designed for the Windows Operating System (OS). Developing this type of software for Windows users, highly increments potential attacks since Windows is by far the most common OS. The Blooover tool runs both in Linux and Windows because it was developed in Java.

To better analyze the impact of BT attacks, their real threats need to be understood. Table III shows information that can be obtained by attacking BT, which can be used for many types of attacks. For example, to perform a BlueBug attack, the BTScanner or the hcitool (a BT configuration utility present in many Linux distributions) can be used to obtain the necessary information from the target devices.

TABLE III. DEVICE INFO OBTAINED BY ATTACK PROCEDURES

| Attack procedure | Device Info | | | | | |
|---|---|---|---|---|---|---|
| | *Address* | *Class* | *Name* | *Type* | *PIN* | *Services* |
| BTScanner | X | X | X | | | |
| hcitool (Linux tool) | X | X | X | | | |
| Blueprinting | X | | | | | |
| BlueScanner | X | | | X | | X |
| BTCrack | | | | | X | |

Table IV shows the impact caused by various attack procedures. Although most of the procedures are directed at a single objective, their inner workings are complex. The following list describes the impacts mentioned in Table IV:

- **SDP**: Allows the discovery of the services enabled and their characteristics.
- **OBEX**: Eases the exchange of binary objects between devices.
- **Security Audits**: Measures technical assessment of a system or application.
- **Send vCards**: Sends messages to other BT devices.
- **Send AT commands**: AT commands are used to control the communication system of the device.
- **DoS attack**: Intends to make the device resources unavailable.
- **Check known vulnerabilities**: Performs an audit on mobile phones to verify whether they are vulnerable to a set of known issues.

The attack impacts shown in Table IV are quite different from each other and it is important to identify which attack is more critical in what concerns the access to private information or full control of the device. For example, the ability of BlueJacking to send text messages to another device seems harmless compared to the ability of BlueBug to send AT commands.

TABLE I. IMPACT OF ATTACK PROCEDURES

| Attack procedure | Impact | | | | | | |
|---|---|---|---|---|---|---|---|
| | ***SDP*** | ***OBEX*** | ***Security audit*** | ***Send vCard*** | ***Send AT commands*** | ***DoS attack*** | ***Check known vulnerabilities*** |
| BTScanner | X | X | | | | | |
| hcitool | X | X | | | | | |
| sdptool | X | | | | | | |
| BlueSnarf | | X | | | | | |
| BlueJacking | | X | | X | | | |
| Blooover | | | X | | | | X |
| BlueBug | | | | | X | | |
| BlueSmack | | | | | | X | |
| BlueSnarf++ | | X | | | | | |
| HeloMoto | | | | X | X | | |

The most relevant attack procedure affecting BT are:

- **BlueSnarf**: Consists on connecting to the OBEX Push Profile (OPP), which allows an easy exchange of files. Since most cases of OPP do not require the service authentication, a weak OBEX implementation may be the entry point for an attacker [2, 15]. If an attacker connects to OBEX and performs an OBEX GET request, files such as the phone book, pictures, or even the calendar can be obtained. More dangerous, is an improper device firmware implementation where an attacker can actually obtain any file, if the name of the file is known.
- **BlueSnarf++**: Is an enhancement of the BlueSnarf, allowing the attacker to also have full read and write access to the device's file system when connected to the OPP [15]. To succeed, this attack requires that the device runs on an OBEX FTP server and can connect to an OBEX Push service without pairing.
- **BlueBug**: Is a name given to a BT vulnerability present on some mobile phones, allowing remote AT commands to be executed on target devices [2, 15]. An attacker exploiting this can obtain information from the mobile phone or even take complete control of the device. This attack can be performed in few seconds, and allows for example, to make a phone call, send and read SMS messages, access and edit the phonebook, forward calls, connect to wireless networks, and change the phone's service provider.
- **BlueJacking**: Consists on sending anonymous vCards (business cards) or text messages to other devices through OBEX, which seems to be physically harmless.
- **HeloMoto**: Is a combination of the BlueSnarf and the BlueBug attacks. The origin of the name is due to a security breach found in some Motorola phones [2, 15].

The two procedures that can actually send AT commands and remotely control the device are BlueBug and HeloMoto. The HeloMoto may not be a widespread attack since it only affects some mobile phones from Motorola. The BlueBug attack seems to be more dangerous since it can be executed on devices from several brands. The brand names are not publicly available because the Trifinite group, who identified this leak, only discloses this information to device manufacturers [15]. Besides the type of information already mentioned that attacks can obtain, access to personal information stored in the mobile phone is also possible. This is a major issue since private data can be traded in the underground market around the globe [16].

In spite of the efforts made to secure and patch BT from the specification, it is still one of the most relevant causes for security problems in mobile phones. Being BT a wireless connection technology, mobile phone users cannot really "see" or "feel" BT and may not be aware of the dangers in case of a security breach. It is estimated that 73% of mobile users are unaware of critical attack types (like BlueSnarf, BlueBug, and Bluejacking) and the damage they may inflict, according to InsightExpress [17, 18].

Moreover, attacks to BT devices can target millions of possible victims. This occurs when the vulnerability affects a widespread device, like the iPhone that accounts for 51.15 million devices worldwide [19]. For this widespread device, a BT vulnerability was discovered in the Service Discovery Profile (SDP), which allows the discovery of enabled services and their characteristics [2]. The attack uses the SDP features to send a maliciously crafted message allowing the attacker to access the root shell of the device.

The discovery of BT security problems is not only of the interest of attackers, but it is also part of the research done for this technology to continue growing as a safe and reliable wireless option. One major player investing their resources to discover BT security failures is The Trifinite Group [15]. Like other common security vulnerabilities (for example, the buffer overflow in desktop and mobile applications), there are many methods and tools to exploit BT devices (like those presented in Table IV).

## V. BLUETOOTH SECURITY PROPOSAL

Given that BT weaknesses are known for some years and BT security has also been improved, we wonder how safe the BT devices are.

To verify the easiness and the assets put in danger by attacking BT we have made the following experiment: we associated an attack machine as a trusted BT device of a mobile phone (this is a standard procedure when connecting together two BT devices) and we tried to execute AT commands. In a real situation this acceptance as a trusted device can be achieved through social engineering, spoofing of other trusted devices, etc. The main idea of this experiment is to expose what an attacker could do with the phone using this simple procedure. Recall that the procedure used is the basis for any of the BT attacks described in the previous section.

### *A. Attack procedure*

The procedure used in our attack was based on the BlueBug attack. The machine used for the attack had Ubuntu Linux 8.04 Operating System installed. As target devices, we used two Nokia mobile phones as shown in Table V. Table V also shows the need to manually accept (or automatic bypass) a connection from a previously known BT device.

TABLE I. MOBILE PHONES ATTACKED

| Mobile phone | Launch year | BT version | Connection bypassed if device paired |
|---|---|---|---|
| Nokia 3110 classic | 2007 | v2.0 | No |
| Nokia 6303i classic | 2010 | v2.1 | Yes |

Before going any further we can see an important difference in the way the phones deal with the pairing of previously associated devices. While the Nokia 3110 classic needs the user to accept (or deny) the incoming BT connection, the Nokia 6303i classic bypasses the authorization and accepts all trusted devices by default. If an

attacker is able to make a rogue device to be associated by the target phone he will be able to attack the phone at any time without being noticed.

The following four steps show the procedure executed from the Linux Shell of the attack machine, as root:

1. **# hcitool scan** – searches for the MAC address of the target BT device.
2. **# rfcomm connect 0 [Target Mac Address] 1** – connects to the target device by RFCOMM.
3. **# minicom –s** configure **A- Serial Device : /dev/rfcomm0** – configures the target device to emulate RS-232 serial ports in order to start a communication.
4. The attacker is ready to send AT commands to the target BT device.

*B. Attack results*

After establishing the RFCOMM to both mobile phones, they were ready to execute the AT commands issued [20]. Table VI shows some AT commands executed, just to have a sample of what was possible to achieve. Recall that AT commands allow executing not only most of the communication functions, but also many other functions to control phone features as the access to the phonebook.

TABLE I. AT COMMANDS EXECUTED DURING THE ATTACK

| AT Command | Description |
|---|---|
| CGMM | Request ME Model Identification |
| CMGF | Message Format |
| CPMS | Preferred Message Storage |
| CPBR | Read Phonebook Entries |

We can see that a trusted BT device can execute a variety of AT commands. It seems that there is no control over it. Our results are corroborated by other studies. To analyze how many BT users could be victim of an attack, a study in London concluded that from 943 mobile phones, 40% had their BT default settings. Moreover, 138 of them were proven to be vulnerable to BlueSnarf attacks [21]. Another test done at the CeBIT technology fair in Hannover concluded that from a range of 1,300 devices, 50 devices were vulnerable to the BlueBug attack [15].

*C. Bluetooth Firewall Proposal*

In our mobile world, BT devices are an easy target for an experienced hacker. Since BT can be used in many daily tasks, it is common practice to have configured in our mobile phone several trusted devices for advertising campaigns [22, 23]. This is a real threat and increases the probability for an attack, if there is no other mechanism to filter BT connections. To reduce the risk of being attacked, users of BT devices should follow best practices, like:

- Turn BT off when not in use.
- Change the default security settings to a more restrict mode whenever possible.
- Remove trusted devices that will not be used.

Not surprisingly, the best protection is turning off BT, but this prevents the use of this wide spread and useful technology. Moreover, all of these practices move the security actions to the user, which is considered by many security practitioners as a bad option. The device should be secure by default, allowing the most important tasks to be done safely, with the least user intervention.

To achieve this kind of filtering and protection on the BT part of the device, we propose the development of a BT Firewall. It could be used to protect against the majority of known attacks, as well as new ones that may appear and use the same entry point. The BT Firewall could as well have a white list and a black list of rules, which can be used to filter devices that should or not be associated with the phone.

The BT Firewall should protect the RFCOMM protocol, which is the second protocol layer on the host side of the BT protocol stack [24], as shown in Fig. 1. By protecting this protocol, all connections that use OBEX, TCP, or intended to send AT commands, could be filtered.

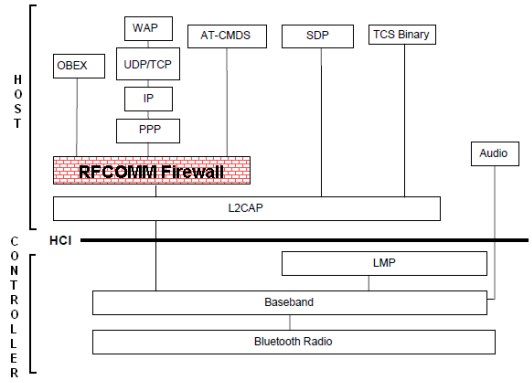



**Figure 1. Bluetooth Protocol Stack with the Firewall**

The proposed BT Firewall may also have the ability to group user profiles into three main categories (Temporary, E-Commerce and @Home, for example), filtering which BT devices have access to its matching profile (Fig. 2).

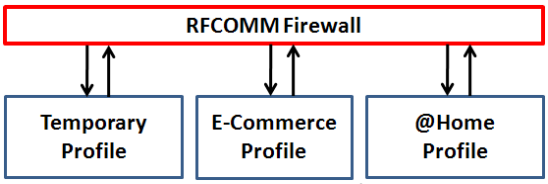



**Figure 2. Main User Profiles**

The @Home profile is for all the devices used in our daily tasks, which should be well known and thus may have a higher level of trust. The Temporary profile is for any type of connection not regularly used and should have a high security restriction. The E-Commerce profile is for BT trading and should have very specific features. We consider that this level of protection in the E-commerce may help potentiate this important streak since this is an area far from being explored yet.

When a new BT device tries to connect, the Firewall will prompt to the user to accept or deny the access, along with the option to associate the connection to a profile. The Firewall will filter MAC addresses, the Universally Unique

Identifier (UUID), and the server channel identifier (in case of a client connection), identifying the BT device to its respective profile. The Firewall may also have a black list of undesired connections.

To achieve security independency between profiles, the same BT device cannot belong to more than one profile. The Firewall will also be responsible for monitoring the traffic and alert the user in case of suspicious actions. BT Firewall filter definition may be updated regularly as soon as new signatures are provided.

The ability of the BT Firewall to authenticate connections by user profiles is a novel approach to protect BT users. Other approaches focus on specific problems, like the protection against malware propagation, using the Blue-Watchdog [25], or improving the already provided encryption of the communication [26]. These approaches are, however limited in the scope, and do not provide the holistic protection that a BT Firewall is capable of.

## VI. CONCLUSIONS

In this paper we analyzed BT security and the most common attack procedures: BlueSnarf, BlueSnarf++, and BlueBug. The BlueSnarf and BlueBug attacks are capable of obtaining private information such as the calendar, the phonebook, and SMS. BlueSnarf++ on the other hand is an enhancement of BlueSnarf with the ability of allowing full read and write access to the file system of the device.

Users of BT enabled devices should follow best practices, like turn off BT when not in use, restrict BT settings, remove trusted devices when no longer needed. However, BT devices should provide by default a safety barrier protecting their users, instead of relying on them to follow the best practices.

In fact, all the attacks affecting BT can be prevented by protecting the device from the RFCOMM protocol. Therefore, we proposed the design a BT Firewall for mobile phones. This feature will filter the RFCOMM connections and associate them by user and profile. By applying this sort of filter at the entry point of the connections, it will prevent BT attacks from being successful. The implementation of this BT Firewall and its evaluation are the tasks we intend to address in future work.

## REFERENCES


[1] Global mobile statistics 2012: http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats#subscribers. Last viewed 23.February.2012.

[2] Terrence OConnor, Douglas Reeves. Bluetooth Network-Based Misuse Detection. NC State University Raleigh. 2008 Annual Computer Security Applications Conference, 2008 IEEE.

[3] Alexander Gostev. Securelist, Bluetooth: London 2006. http://www.securelist.com/en/analysis?pubid=188833782. Last viewed 23.February.2012.

[4] BLUETOOTH SPECIFICATION Version 4.0 [Vol 0]. Master Table of Contents & Compliance Requirements. Publication date: 17 December 2009.

[5] Bluetooth Special Interest Group (SIG): https://www.bluetooth.org/. Last viewed 23.February.2012.

[6] Karen Scarfone, John Padgette. Guide to Bluetooth Security. National Institute of Standards and Technology. Special Publication 800-121. September 2008.

[7] A Gartner, Press Releases. Egham, UK, February 23, 2010. http://www.gartner.com/it/page.jsp?id=1306513. Last viewed 23.February.2012.

[8] Bluetooth: one billion devices and growing http://www.macworld.co.uk/digitallifestyle/news/?newsid=16477 Last viewed 17.February.2012.

[9] Bluetooth: 2 Billion In 2013 http://hothardware.com/News/InStat-Predicts-BluetoothEnabled-Device-Shipments-Will-Top-2-Billion-In-2013/. Last viewed 06.April.2012.

[10] Catherine Roseberry, About.com: http://mobileoffice.about.com/cs/traveladvice/qt/usingcellphone.htm. Last viewed 23.February.2012.

[11] Wireless and Mobile News. Mobile Apps Race to Serve Auto Market @IAA. 12.October.2009: http://www.wirelessandmobilenews.com/2009/10/mobile-apps-race-to-serve-auto-market-iaa-says-isuppli.html. Last viewed 23.February.2012

[12] Parrot, wireless devices for mobile phones: http://www.parrot.com/usa/. Last viewed 23.February.2012.

[13] Media Ford. Ford Teams up with Microsoft to deliver SYNC; In-car Digital System Exclusive to Ford. 7.January.2007: http://media.ford.com/Article_Display.Cfm?Article_Id=25168. Last viewed 23.February.2012.

[14] Andrew Y. Lindell. Attacks on the Pairing Protocol of Bluetooth v2.1. Aladdin Knowledge Systems and Bar-Ilan University, Israel. June 25, 2008.

[15] Trifinite Group: http://trifinite.org/trifinite_stuff.html. Last viewed 23.February.2012.

[16] Symantec. Symantec Report on the Underground Economy. Published November 2008.

[17] Lynn Tan, ZDNet. Symantec warns users over Bluetooth security. 21.September.2007: http://www.zdnet.com/news/symantec-warns-users-over-bluetooth-security/165841. Last viewed 23.February.2012.

[18] Don Reisinger, CNET. Bluejacking, bluesnarfing and other mobile woes. 22.August.2007: http://news.cnet.com/8301-13506_3-9764450-17.html. Last viewed 23.February.2012.

[19] Greg Kumparak, Mobile Crunch. Apple sold 8.75 million iPhones last quarter, 51.15 million since launch. 20.April.2010: http://www.mobilecrunch.com/2010/04/20/apple-q2-earnings-million-iphones. Last viewed 23.February.2012.

[20] NOKIA 30 GSM CONNECTIVITY TERMINAL AT COMMAND GUIDE, Issue 2.0. Copyright © Nokia 2002.

[21] Kevin Streff, Justin Haar. An Examination of Information Security in Mobile Banking Architectures. Dakota State University. Journal of Information Systems Applied Research. June 10, 2009.

[22] Proximity Marketing: http://www.bluetoothmarketing.com/. Last viewed 23.February.2012.

[23] Marketing Platform: http://www.breeze-tech.co.uk/. Last viewed 23.February.2012.

[24] JavaTM APIs for BluetoothTM Wireless Technology (JSR-82). Motorola Wireless Software, Applications & Services. 1.0a, April 5, 2002.

[25] Mohamed GHALLALI, Driss EL OUADGHIRI, Mohammad ESSAAIDI, Mohamed BOULMALF. Mobile Phones Security: The Spread of Malware via MMS and Bluetooth, Prevention Methods. MoMM2011, 5-7 December, 2011, Ho Chi Minh City, Vietnam.

[26] Yu Xin, Yan Ting. A Security Architecture Based on User Authentication of Bluetooth. College of Automation, Beijing Union University, Beijing 100101, China. © 2009 IEEE.

# ANEXO B – POSTER INFORUM 2012

Este anexo apresenta o *poster* científico *Bluetooth security analysis for mobile phones* e que foi apresentado no INForum 2012 na Universidade NOVA, Lisboa, a 7 de setembro de 2012.

## Bluetooth Security Analysis for Mobile Phones

João Alfaiate, José Fonseca, *Guarda Polytechnic Institute, Portugal*

### Objective

Study Bluetooth (BT) security problems in mobile devices, analyze common BT mobile attacks, and how to prevent them. We paid special attention to the most critical security problems affecting BT, which are **BlueSnarf**, **BlueSnarf++** and **BlueBug**.

### Bluetooth Evolution

Since the first public version of BT in 1999, five updated versions were released until 2010. They are shown in Table I, along with the most relevant features for our study.

| BT Versions | Year | Data Rate (Mbps) | Faster connection | SSP | Security Mode 4 | Bug fixes | Error detection | Synchronization | Data Rate | L2CAP | HCI for AMP | Security for AMP | Power consumption |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1.1 | 2002 | | | | | X | | | | | | | |
| 1.2 | 2003 | 1 | | | | | X | X | | | | | |
| 2.0 | 2004 | 3 | X | | | | | | X | | | | |
| 2.1 | 2007 | 3 | | X | X | | | | | | | | X |
| 3.0 | 2009 | 24 | | | | | | | X | X | X | X | X |
| 4.0 | 2010 | 24 | | | | | | | | | | | X |
| | | | New Features | | | Enhancement Features | | | | | | | |

**Table 1. Bluetooth Features**

### Bluetooth Attack Analysis

During the research, we found 6 tools and 11 methods to attack BT. The majority of the attacks appeared between years 2004 and 2007. This timeframe corresponds to the upgrade of BT from version 2.0 to 2.1. Figure 1 shows the most dangerous attacks. The following is their definition:

- **BlueSnarf**: Consists on connecting to the OBEX Push Profile (OPP).
- **BlueSnarf++**: Is an enhancement of the BlueSnarf.
- **BlueBug**: A BT vulnerability present on some mobile phones, allowing remote AT commands to be executed.

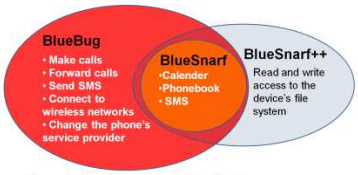



**Figure 1. Most dangerous BT attacks**

### Experiment done Attacking BT Devices

The procedure used in our attack was based on the BlueBug attack. As target devices, two Nokia mobile phones were used as shown in Figure 2. While the Nokia 3110 classic needs the user to accept (or deny) the incoming BT connection, the Nokia 6303i classic bypasses the authorization and accepts all trusted devices by default.

After establishing the RFCOMM connection to both mobile phones, they were ready to execute AT commands. In this case we can see that Version 2 seems more secure than Version 2.1.

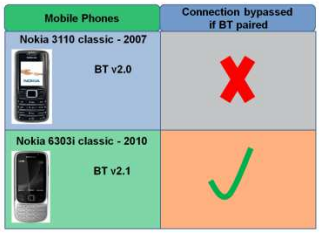



**Figure 2. Most dangerous attacks**

### Firewall Proposal

BT devices are an easy target for an experienced hacker. Since BT can be used in many daily tasks, it is common practice to have configured in our mobile phone several trusted devices.

To achieve protection on the BT device, we propose the development of a BT Firewall. It could be used to protect against the majority of known attacks, as well as new ones that may appear and use the same entry point. The BT Firewall could as well have a white list and a black list of rules, which can be used to filter devices that should or not be associated with the phone.

The BT Firewall should protect the RFCOMM protocol, which is the second protocol layer on the host side of the BT protocol stack (Figure 3). By protecting this protocol, all connections that use OBEX, TCP, or intended to send AT commands, can be filtered.,

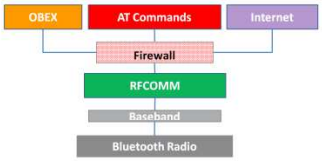



**Figure 3. Firewall in the BT Protocol Stack**

### User Profiles

The proposed BT Firewall may also have the ability to group user profiles into three main categories (@Home , Temporary, and E-Commerce), filtering which BT devices have access to its matching profile (Figure 4).


**Figure 4. Main User Profiles**

### Conclusions

The Firewall will filter the RFCOMM connections and associate them by user and profile. By applying this sort of filter at the entry point of the connections, it will prevent BT attacks from being successful.

# ANEXO C - LISTA DE COMANDOS AT

Os comandos AT são utilizados para controlar *modems* e vêm de comandos Hayes que foram usados pelos modems inteligentes Hayes. Os comandos Hayes começavam com as letras AT (que é a abreviatura de *Attention*) para indicar a atenção do MODEM [*AT Commands*].

Ao testar diversos comandos AT no telemóvel *Nokia 6303i classic* (telemóvel usado na experiência, que consta no capitulo 3, e que permitia a ligação através do RFCOMM sem ser detetado), verificou-se que muitos comandos não foram capazes de ser executados. A Tabela 17 mostra alguns dos comandos executados e o seu resultado. Estes comandos foram selecionados pela sua diversidade e pelo facto de serem procedimentos usados em ataques como o *Bluebug* e o *BlueSnarf*.

**Tabela 17.** Comandos AT [*NOKIA AT Commands*, 2000]

| Comando | Descrição | Resultado |
|---|---|---|
| AT+CGMM | Request Model ID | Bem sucedido |
| AT+CMGF | Set Message Format | Bem sucedido |
| AT+CPMS | Preferred Message Storage | Bem sucedido |
| AT+CMGL | List Messages | Erro |
| AT+CMGW | Write Message to Memory | Erro |
| AT+CMSS | Send Message from Storage | Erro |
| AT+CMGS | Send Message | Erro |
| AT+CPBR | Read Phone Book Entry | Bem sucedido |
| AT+CPBW | Write Phone Book Entry | Erro |
| ATD | Dial Command | Erro |
| ATA | Answer Command | Erro |
| AT+CMGR | Read Message | Erro |
| AT+CNMI | New Message Indications to DTE | Erro |
| AT+CGSN | Request Product Serial Number Identification | Bem sucedido |
| AT+CGMR | Request Revision Identification | Bem sucedido |
| AT+CPBS | Select Phone Book Memory Storage | Bem sucedido |

# ANEXO D – EXEMPLO DE LIGAÇÕES RFCOMM

Este anexo fornece um exemplo de código que poderá ser útil na elaboração de *software* que envolve o Bluetooth. Este exemplo mostra como é aceite as ligações de RFCOMM, retorna o que é recebido por parte do cliente e exibe as mensagens do cliente bem como a *string* da ligação utlizada para se ligar ao serviço.

Código retirado de [*C Bala Kumar et al.*, Motorola 2003]

```
/*
 * C Bala Kumar, Paul J. Kline & Timothy J. Thompson,
 * Bluetooth Application Programming with the Java APIs
 * Published By Morgan Kaufmann Publishers
 * (c) Motorola, Inc. 2003.
 *
 * Permission to use, copy, modify, and distribute this
 * software and its documentation for NON-COMMERCIAL purposes
 * and without fee is hereby granted provided that this
 * copyright notice appears in all copies.
 *
 * THE AUTHORS AND PUBLISHER MAKE NO REPRESENTATIONS OR
 * WARRANTIES ABOUT THE SUITABILITY OF THE SOFTWARE, EITHER
 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
 * PARTICULAR PURPOSE, OR NON-INFRINGEMENT. THE AUTHORS
 * AND PUBLISHER SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED
 * BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING
 * THIS SOFTWARE OR ITS DERIVATIVES.
 */

package com.jabwt.book;

import java.lang.*;
import java.io.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;
import javax.bluetooth.*;

public class EchoServer extends BluetoothMIDlet {
    /**
     * Adds the connection string to use to connect to
     * this service to the screen.
     *
     * @param f the Form to add the connection string to
     * @param notifier the notifier object to retrieve
     * the connection
     * string from
     */
    private void displayConnectionString(Form f,
        StreamConnectionNotifier notifier) {

        try {
            // Retrieve the connection string to use to
            // connect to this server
```

```
            LocalDevice device = LocalDevice.getLocalDevice();
            ServiceRecord record = device.getRecord(notifier);

            String connString = record.getConnectionURL(
                ServiceRecord.NOAUTHENTICATE_NOENCRYPT, false);

            int index = connString.indexOf(';');
            connString = connString.substring(0, index);

            // Display the connection string on the Form
            f.append("Connection String:\n");
            f.append(connString);
            f.append("\n");
        } catch (BluetoothStateException e) {
            f.append("BluetoothStateException: " +
            e.getMessage());
        }
    }

    /**
     * Accepts connections from RFCOMM clients and
     * echoes back what is received from the client. This
     * method also displays the messages from a client
     * on a Form. It also displays on the Form the
     * connection string to use to connect to this service.
     */
    public void run() {

        // Create the output Form and set it to be the
        // current Displayable
        Form msgForm = new Form("Echo Server");
        msgForm.addCommand(new Command("Exit",
            Command.EXIT, 1));
        msgForm.setCommandListener(this);
        Display.getDisplay(this).setCurrent(msgForm);

        try {
            //Create the notifier object
            StreamConnectionNotifier notifier =
                (StreamConnectionNotifier)
                Connector.open(
                "btspp://localhost:123456789ABCDE"
                + "name=Echo Server");

            // Display the connection string on the Form
            displayConnectionString(msgForm, notifier);

            // Continue accepting connections until the MIDlet
            // is destroyed
            for (;;) {
                StreamConnection conn = notifier.acceptAndOpen();
                OutputStream output = conn.openOutputStream();
                InputStream input = conn.openInputStream();

                // Continue reading the input stream until the
                // stream is closed. Display the data on the
                // screen and write it to the output stream.
                byte[] data = new byte[10];
                int length = 0;
                while ((length = input.read(data)) != -1) {
                    msgForm.append(new String(data, 0, length));
                    output.write(data, 0, length);
                }
```

```
                // Close the streams and the connection
                output.close();
                input.close();
                conn.close();
            }
        } catch (IOException e) {
            msgForm.append("IOException: " + e.getMessage());
        }
    }
}
```